

Faculdade de Engenharia da Universidade do Porto



**Desenvolvimento de programa educativo para o
ensino de semicondutores**

Emanuel Alexandre da Silva Fontes

Versão Final

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major Automação

Orientador: Prof. Doutor Armando Araújo
Co-orientador: Prof. Doutor Rui Chibante

Março de 2010

Resumo

No mundo da electrónica, os componentes baseados em materiais semicondutores revelam a sua importância em aplicações como rectificadores ou conversores, em que díodos e transístores são os seus principais constituintes. Dado o seu interesse e complexidade, o ensino e aprendizagem destas matérias, no meio académico, surge como uma tarefa árdua e morosa, tornando-se fundamental encontrar soluções que auxiliem tanto os docentes como os estudantes. Neste sentido, foi proposto o desenvolvimento de um programa educativo, de simulação de díodos e transístores, inseridos em circuitos típicos de aplicação, que permitisse, aos estudantes, analisar e compreender a influência dos vários factores envolvidos (parâmetros, topologia e elementos de circuito) no seu funcionamento.

O desenvolvimento da presente dissertação passou por várias etapas descritas do modo seguinte:

Em primeiro lugar relata-se, de forma sucinta, a origem, evolução e estado actual dos dispositivos semicondutores (Díodos e Transístores).

De seguida, apresentam-se duas aplicações educativas, desenvolvidas para o apoio ao ensino das matérias em causa, e, por fim, uma ferramenta de simulação de circuitos eléctricos, utilizada em larga escala em meios académicos.

Posteriormente, é feito um estudo sumário dos modelos dos semicondutores utilizados na simulação dos circuitos. Estes modelos são apresentados na bibliografia estudada e caracterizam-se por apresentarem um comportamento híbrido, utilizando formulações baseadas nas características físicas dos semicondutores, para as zonas pouco dopadas, responsáveis pela dinâmica do dispositivo, complementadas com aproximações empíricas associadas a outras zonas semicondutoras (junções, zonas de base, etc)

Uma vez que o trabalho desenvolvido passou pela implementação de uma interface gráfica, surgiu assim a necessidade de fazer um levantamento das ferramentas de criação de interfaces gráficas, disponíveis no mercado, que melhor se adaptavam aos objectivos propostos. Depois de escolhida a ferramenta que cumprisse os requisitos (Borland Delphi 7), fez-se um estudo do simulador (IsSpice - ICAPS) com o qual é possível simular circuitos utilizando os modelos (Híbridos) desenvolvidos para os semicondutores .

No que diz respeito à implementação da interface propriamente dita, introduziu-se o conceito de programação com classes e programação orientada a objectos, sendo que todo o desenvolvimento da interface se baseia nestes conceitos. Com o intuito de clarificar o funcionamento da interface utilizaram-se diagramas UML (*Unified Modeling Language*).

Por último é feita a validação da interface desenvolvida e apresentam-se sugestões futuras para melhorar o seu funcionamento.

Abstract

In terms of electronic circuits, the basic components of a rectifier or converter circuit are diode and transistor, which reveal extreme importance due to their properties in terms of conducting current. Teaching this subject in a classroom can be difficult to students and teachers and it can become boring. Therefore, it's essential find solutions in order to help both teachers and students.

Therefore, it was proposed to develop educational software to simulate circuits based on diodes and transistors that allows students to analyze and understand precisely the influence of its physical parameters in dynamic operation.

The development of this dissertation passes through several stages. In the first stage a brief study of origins, evolution and terms of conducting of semiconductors has been done. Then it will be presented two educational software's developed to support the teaching of these subjects and, at last, one of the most used simulator tool in classrooms.

Subsequently, a brief study about semiconductors models will be made. These models, which are presented in the studied biography, are known for the hybrid behavior, based on empirical formulations and physical specifications of semiconductors.

Since the main objective is to create a graphical interface, a brief survey of tools was made, in order to select the best existing graphical interfaces, which best fits the proposed objectives. After choosing the indicated tool (Borland Delphi 7), a study about the circuit simulator (IsSpice - ICAPS) that enables circuit simulations based on the developed models described above will be made.

Regarding to the interface implementation itself, it was introduced the concept of classes and - object oriented programming -, because the entire development of the interface is based on these concepts. In order to clarify the use to of interface, several UML (Unified Modeling Language) diagrams were used, such as Class Diagram and Use to Diagram. To summarize, the developed software will be validated, and some suggestions will be presented for further improvements.

Agradecimentos

Gostaria de agradecer ao meu orientador, Professor Doutor Armando Araújo pela ajuda e conselhos durante o desenvolvimento desta dissertação.

Ao Professor Doutor Rui Chibante, co-orientador, pela sua disponibilidade e auxílio durante o desenvolvimento do trabalho e na revisão do documento produzido.

Aos meus amigos que passaram o semestre a elaborar um trabalho semelhante de dissertação mas que, sempre que necessário, deram a sua opinião, ajuda e principalmente motivação quando o caminho se tornou por vezes mais sinuoso.

Um agradecimento especial ao meu fiel amigo, o Engenheiro André Levi por todo o apoio fornecido ao longo destes últimos meses, nomeadamente na revisão ortográfica da dissertação.

À Cristiana, a minha namorada, pela atenção, carinho, afecto, motivação e apoio fornecido ao longo do último ano e meio.

Ao meu irmão pelas constantes dúvidas colocadas sobre o trabalho desenvolvido, o que permitiu ter uma noção das dificuldades dos estudantes mais novos na aprendizagem de novos conceitos.

Por último, mas em primeiro plano, gostaria de agradecer de forma incansável aos meus pais pelo facto de terem investido e acreditado sempre em mim.

Índice

Resumo	i
Abstract.....	iii
Agradecimentos	v
Índice.....	vii
Lista de Figuras	xi
Lista de Tabelas	xv
Abreviaturas e Símbolos	xvii
Capítulo 1	1
1.1 - Enquadramento da dissertação.....	1
1.2 - Objectivos.....	2
1.3 - Requisitos	2
1.4 - Estrutura do documento	3
Capítulo 2	5
Estado da arte sobre aplicações de simulação de circuitos	5
2.1 - Introdução.....	5
2.2 - Díodos e transístores - um pouco de História	6
2.3 - Aplicações educacionais	24
2.3.1 - TESE - Teaching of Semiconductors	24
2.3.2 - SEMISOFT - Semiconductor Programa	25
2.4 - O simulador PSIM®	31
2.5 - Conclusões	33
Capítulo 3	35
Modelos dos semicondutores em estudo	35
3.1 - Introdução.....	35
3.2 - Síntese dos modelos dos semicondutores utilizados	35
3.3 - Circuitos de Teste	37
3.3.1 - Díodo	37
3.3.2 - IGBT.....	39
3.4 - Conclusões	41
Capítulo 4	43

Ferramentas de programação utilizadas.....	43
4.1 - Introdução	43
4.2 - A ferramenta de simulação ICAPS®	44
4.2.1 - Esquemático.....	44
4.2.2 - Editor de texto.....	45
4.2.3 - Simulador	46
4.2.4 - Visualizador de Sinais.....	47
4.3 - Ferramenta de Programação (Delphi 7)	48
4.3.1 - Tomada de decisão	48
4.3.2 - Linguagem de programação	49
4.3.3 - Exemplo de aplicação	50
4.4 - Conclusões	52
Capítulo 5	53
Implementação da interface de simulação	53
5.1 - Introdução	53
5.2 - Regras básicas para uma interface de sucesso	54
5.3 - Implementação com Classes.....	56
5.4 - Definição das classes na interface.....	59
1. Classe UComponentes	59
2. Classe UDiodo	60
3. Classe UBobine	60
4. Classe UFontes	61
5. Classe UResistência	61
6. Classe UFontes_Comando.....	62
7. Classe UTransistor.....	62
8. Classe UReferência.....	63
9. Classe UCircuito.....	63
10. Classe UGraficos_2D	64
11. Classe Ugraficos_3D.....	65
12. Classe UEscolha_Graficos	66
13. Classe UExplicacao_teorica	67
14. Classe UPainel.....	67
15. Classe UPainel_Diodos.....	68
16. Classe UPainel_Transistor	69
17. Classe UPainel_Fontes_Comando.....	70
18. Classe Upainel_Marcadores	70
19. Classe UPainel_Zoom	71
20. Classe UErro	71
21. Classe UMain.....	72
22. Unit funções	72
5.5 - Diagramas UML	73
5.5.1 - UML Diagrama de Classes	73
5.5.2 - UML Casos de Uso	74
5.6 - Funções importantes	78
5.7 - Representação gráfica	87
5.8 - Instalação de um único ficheiro.....	87
5.9 - Gravação para PDF	88
5.10 - Conclusões.....	88
Capítulo 6	89
Demonstração e Validação	89
6.1 - Introdução	89
6.2 - Demonstração	89
6.3 - Validação.....	101
6.4 - Conclusões	102

Capítulo 7	103
Conclusões e Trabalho Futuro	103
7.1 - Conclusões	103
7.2 - Trabalho Futuro	104
Referências	105
Anexo A.....	109

Lista de Figuras

Figura 2-1- Esquema da lâmpada de Edison [30].	7
Figura 2-2 - Esquema da Lâmpada de Edison com a introdução da placa de metal [30].	7
Figura 2-3- Rectificador de “Bigodes de Gato”, por Braun [15].	8
Figura 2-4- Efeito de Hall. 1- Carga sofre um desvio provocado pelo campo magnético, 2- Campo magnético, 3- Cargas, 4-Imans.	9
Figura 2-5- Válvula de Fleming [22].	10
Figura 2-6- a)-Efeito de Edison, b)- Válvula de Fleming, c)- Tríodo de Lee de Forrest [31]. ..	11
Figura 2-7- Diagrama temporal da evolução dos dispositivos electrónicos constituídos por semicondutores antes da Segunda Guerra Mundial.....	13
Figura 2-8 - Transístor desenvolvido por John Bardeen e Walter Brattain [32].	14
Figura 2-9 - Esquema do primeiro transístor [17].	15
Figura 2-10- Esquema do primeiro transístor bipolar de junção [17].	16
Figura 2-11- Transístor de junção bipolar [17]	16
Figura 2-12 - Primeiro circuito integrado desenvolvido por Jack Kilby´s [17].	17
Figura 2-13- Primeiro CI desenvolvido pela Fairchild através de um processo planar [17].	18
Figura 2-14- Lei de Moore [33].	19
Figura 2-15 - Esquema do díodo molecular desenvolvido na Universidade do Estado do Arizona [36].	20
Figura 2-16- Visualização da variação da estrutura do benzeno quando se aplica uma tensão aos eléctrodos de ouro [37].	21
Figura 2-17- Estrutura do grafeno [38]	21
Figura 2-18- Diagrama temporal da evolução dos dispositivos electrónicos constituídos por materiais semicondutores depois da Segunda Guerra Mundial.	23

Figura 2-19 - Possibilidade de definir valores para o componente seleccionado [3].	25
Figura 2-20- Menu Início do programa SEMISOFT.	26
Figura 2-21 - Menu de simulação, SEMISOFT.	26
Figura 2-22 - Simulação de circuito com díodo.	27
Figura 2-23 - Janela de parâmetros do díodo, circuito e simulação.	28
Figura 2-24 - Gráficos a duas dimensões da tensão e corrente no díodo.	29
Figura 2-25 - Representação a três dimensões -SEMISOFT.	30
Figura 2-26 - Menu de configuração dos caminhos do sistema -SEMISOFT.	30
Figura 2-27 - Ambiente de desenho do PSIM®. 1- Barra de atalhos; 2-Barra de atalho dos componentes mais utilizados; 3- Desenho do circuito.	31
Figura 2-28 - Janela para modificar características do componente (díodo).	32
Figura 2-29 - Janela de visualização dos dados. 1- Barra de atalhos, 2- Barra de máximos e mínimos locais e globais, 3- Visualização dos gráficos.	33
Figura 3-1 - Esquema de um díodo convencional.	36
Figura 3-2 - Estrutura típica de um díodo de potência.	36
Figura 3-3 - Estrutura típica de um IGBT.	36
Figura 3-4 - RC para a solução da EDA para elementos lineares [6].	37
Figura 3-5 - Circuito de teste do díodo.	38
Figura 3-6 - Circuito de teste do IGBT (Tteste).	39
Figura 4-1- Exemplo do desenho de um circuito no ICAPS.	44
Figura 4-2- Exemplo da edição de um circuito, utilizando o módulo de editor de texto, IsEd4.	45
Figura 4-3 - Janela de Simulação “IsSpice4”. 1 - “Simulation Status” ; 2 - “Errors and Status”, 3 - “Output”, 4 - “Simulation Control”.	46
Figura 4-4 - Módulo de visualização dos sinais - IntuScope. 1 - “Comand”, 2 - “Waveform Calculator”, 3 - “Scalling”, 4 - “Output Record”, 5 - “Graph”.	47
Figura 4-5 - Exemplo ilustrativo de um pequeno programa “Hello World”. 1 - Janela de atalhos. 2 - “Object TreeView”. 3 - “Object Inspector”. 4 - Janela de programação. 5 - “Form”.	50
Figura 4-6 - Janela do exemplo apresentado em execução.	51
Figura 4-7 - 1- No caso de o “Radio Button” não estar seleccionado. 2- No caso de o “Radio Button” estar seleccionado.	52
Figura 5-1 - Implementação da Classe Carro.	58
Figura 5-2 - Classe UComponentes 1- Nome da Classe; 2- Atributos; 3- Métodos.	60

Figura 5-3 - Classe UDíodo. 1-Nome da Classe; 2- Atributos; 3- Métodos.....	60
Figura 5-4 - Classe UBobine. 1- Nome, 2- Atributos, 3-Métodos.....	61
Figura 5-5 - UFontes. 1- Nome, 2- Atributos, 3- Métodos.....	61
Figura 5-6 - Classe UResistência. 1- Nome, 2- Atributos, 3- Métodos.	61
Figura 5-7 - Classe UFontes_Comando. 1- Nome, 2- Atributos, 3- Métodos.	62
Figura 5-8 - Classe UTransistor. 1- Nome, 2- Atributos, 3- Métodos.....	63
Figura 5-9 - Classe UReferência.1- Nome, 2- Atributos, 3-Métodos.	63
Figura 5-10 - Classe UCircuito. 1- Nome, 2- Atributos, 2a- Instâncias dos vários componentes, 3- Métodos.	64
Figura 5-11 - Classe UGraficos_2d.1- Nome, 2- Atributos gráficos (“Form”), 3- Atributos, 4- Métodos, 5- Métodos gráficos (eventos).	65
Figura 5-12 - Classe Ugraficos_3D. 1-Nome, 2-Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos Gráficos.....	66
Figura 5-13 - Classe UEscolha_graficos. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos Gráficos.....	67
Figura 5-14 - Classe UExplicação_teorica. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.	67
Figura 5-15 - Classe UPainel. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.....	68
Figura 5-16 - Classe UPainel_Díodos. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.	69
Figura 5-17 - Classe Upainel_transistor. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.	69
Figura 5-18 - Classe Upainel_fontes_comando. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.....	70
Figura 5-19 - Classe Upainel_Marcadores. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.	71
Figura 5-20 - Classe UPainel_Zoom. . 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.	71
Figura 5-21 - Classe UErro. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.....	72
Figura 5-22 - Classe UMain. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.....	72
Figura 5-23 - Unit Funções.	73
Figura 5-24 - Diagrama UML de Classes da interface desenvolvida.	74
Figura 5-25 - Diagrama UML Casos de Uso.	75

Figura 5-26 - Trecho de código do ficheiro Diodo.ckt. 1- Parâmetros do Díodo.....	79
Figura 5-27 - Fluxograma da função “Proc_e_subs”.....	80
Figura 5-28- Trecho do ficheiro “Diodo.ckt”.1- Bobine, 2- Fonte de Corrente.....	81
Figura 5-29 - Fluxograma da função proc_e_subs_pack.	82
Figura 5-30 - Fluxograma da função “Simulação”.	83
Figura 5-31 - Trecho do ficheiro Diodo.out.	84
Figura 5-32 - Trecho2 do ficheiro diodo.out.	85
Figura 5-33 - Fluxograma da função “Carrega_valores_saída”.	86
Figura 6-1 - SWEESC - Janela de arranque.	89
Figura 6-2 - SWEESC - Janela de apresentação.	90
Figura 6-3 - SWEESC - Circuito do díodo.	91
Figura 6-4 - SWEESC - Legenda do circuito.....	91
Figura 6-5 - SWEESC - Circuito do díodo - Legenda dos componentes - V1.....	92
Figura 6-6 - SWEESC - Circuito do díodo - Legenda dos componentes - DUT.....	92
Figura 6-7 - SWEESC - Circuito do díodo - Painel de alteração de valores.	93
Figura 6-8 - SWEESC - Simulação de um circuito.	93
Figura 6-9 - SWEESC - Simulação do díodo - Janela de confirmação de simulação bem sucedida.	94
Figura 6-10 - SWEESC- Janela de escolha dos gráficos a simular.	95
Figura 6-11 - SWEESC - Janela de visualização dos gráficos 2D.	96
Figura 6-12 - SWEESC- Janela de visualização dos gráficos 2D (2 simulações).....	96
Figura 6-13 - SWEESC - Painel zoom.....	97
Figura 6-14 - SWEESC - Painel Marcadores.	97
Figura 6-15 - SWEESC - Gráficos 2D com utilização dos painéis auxiliares.	98
Figura 6-16 - SWEESC - Escolha da simulação para gerar Gráfico 3D.	98
Figura 6-17 - SWEESC - Visualização tridimensional.	99
Figura 6-18 - SWEESC - Janela de visualização dos caminhos do sistema.	100
Figura 6-19 - SWEESC - Janela de escolha do caminho.	100
Figura 6-20 - Validação da interface - Variação do tempo médio de vida de um díodo.	101
Figura 6-21 - SWEESC - Explicação teórica sobre a variação do tempo médio de vida.	102

Lista de Tabelas

Tabela 3-1 - Parâmetros do modelo do díodo.	38
Tabela 3-2- Parâmetros do modelo IGBT.	40
Tabela 4-1 - Critérios de selecção da ferramenta gráfica a utilizar.	49
Tabela 5-1 - Descrição do caso de uso “Simular Circuito”.	76
Tabela 5-2 - Descrição do caso de uso Simulações Pré-Definidas.	76
Tabela 5-3 - Descrição do caso de uso Alterar Pasta Gravação.	77
Tabela 5-4 - Descrição do caso de uso Gravar no formato .pdf	77
Tabela 6-1 Tabela da variação do tempo médio de vida.	101

Abreviaturas e Símbolos

Lista de abreviaturas (ordenadas por ordem alfabética)

CI	Circuito Integrado
FET	Field Effect Transistor
IGBT	Insulated Gate Bipolar Transistor
MOS	Metal Oxide Semiconductor
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
NI	National Instruments
NPN	Negative, Positive, Negative
PN	Positive, Negative
PSIM	Power Simulation
Semisoft	Semiconductor Software
SGM	Segunda Guerra Mundial
SO	Sistema Operativo
SWEESC	Software Educacional Ensino Semicondutores
TBJ	Transistor Bipolar Junção
TESE	Teaching Of Semiconductors
TI	Texas Instruments
UML	<i>Unified Modeling Language</i>

Lista de símbolos

V1- tensão de Alimentação
L - Indutância de fugas 1
LF - Indutância de fugas 2
L1 - Indutância principal
DUT - Díodo de Teste

Pulse - Fonte de Comando
A - Área Total
Wb - Comprimento da base
Nb - Concentração por volume da base
hp - Parâmetro de recombinação
 τ - Tempo médio de vida
Vcc - Tensão de alimentação do circuito do IGBT
Vgg - Tensão de controlo
Lc - Indutância de fugas
Rc - Resistência de carga
Rg - Resistência da Gate
Agd - Área gate-dreno
Cgs - Capacidade gate-fonte
Coxd - Capacidade de depleção óxido gate-dreno
Kf - Factor de transcondutância da região de triodo MOS
Kp - Transcondutância da região de saturação MOS
Vth - Tensão limiar de condução

Capítulo 1

Introdução

1.1 - Enquadramento da dissertação

No meio académico o ensino dos semicondutores, e de toda a teoria que lhe está associada, surge como uma tarefa árdua e morosa, capaz de provocar algum desinteresse e desmotivação nos estudantes aquando da sua aprendizagem.

A utilização de ferramentas computacionais de simulação de circuitos em meios académicos não é um tema devidamente explorado [1][2][3]. Tradicionalmente, utilizam-se simuladores comerciais que, ou são extremamente complexos, ou fazem uso de modelos de semicondutores fundamentalmente empíricos, com carências ao nível de explicações teóricas dos fenómenos eléctricos que ocorrem, o que acaba por provocar alguma desmotivação nos estudantes nesta área de aprendizagem.

Assim, torna-se crucial contrariar esta tendência, sendo essa a principal motivação da presente dissertação. Para isso, propôs-se o desenvolvimento de uma interface computacional de simulação capaz de permitir, ao utilizador, analisar e compreender com precisão a influência dos parâmetros físicos dos semicondutores (diodos e transístores) e do circuito, em regime dinâmico de funcionamento. Tendo em vista a consecução desse objectivo a aplicação deverá fazer uso de modelos físicos de semicondutores, permitindo visualizar as formas de onda características (tensões e correntes, a distribuição dos portadores de carga nas zonas pouco dopadas, bem como as explicações teóricas dos principais fenómenos que ocorrem em cada um dos circuitos analisados.

Desta forma, a dedicação, o tempo, o esforço e o investimento dispensados no estudo e implementação de um conversor electrónico, por mais simples que seja a sua configuração,

podem ser reduzidos recorrendo a ferramentas computacionais que, de uma forma simples e eficaz, permitam simular o funcionamento dos semicondutores, em função de diversos parâmetros dos circuitos de carga e de comando.

1.2 - Objectivos

Os objectivos desta dissertação passam por desenvolver uma interface de simulação capaz de estimular a aprendizagem dos princípios físicos associados aos semicondutores (diodos e transístores), realizar um estudo sucinto dos modelos dos semicondutores utilizados e do estado da arte dos assuntos associados ao tema da presente dissertação.

1.3 - Requisitos

A interface desenvolvida deve cumprir diversos requisitos de forma a poder ser utilizada no ensino. Estes requisitos passam por aspectos como a facilidade em operar com a interface gráfica, a compatibilidade da aplicação com o Sistema Operativo (SO) e a possibilidade de guardar todas as simulações realizadas.

Ao nível da componente visual, esta deve ser intuitiva, organizada e de fácil manuseamento de forma a focalizar a atenção do estudante na simulação e na compreensão dos fenómenos ocorridos.

A interface deve permitir que o utilizador simule circuitos pré-definidos com os parâmetros de simulação, carga e de comando que pretender. Os resultados devem ser apresentados em gráficos a duas e a três dimensões e a informação de cada simulação gravada em ficheiros, criando assim um histórico de simulações.

No que diz respeito à instalação, a ferramenta deve-se adaptar a qualquer SO recente da Microsoft (Windows XP, Windows Vista, Windows 7).

Ao nível do histórico de simulações, a interface deve permitir que o utilizador possa a qualquer momento redefinir o caminho de gravação do histórico, possibilitando a utilização de discos de armazenamento pessoais.

As representações gráficas bidimensionais devem apresentar a evolução temporal da tensão e da corrente, no semiconductor em análise, em função dos parâmetros de simulação de carga e comando definidos. A interface deve permitir que o utilizador efectue várias simulações distintas do mesmo circuito, alterando apenas um determinado parâmetro, facilitando, assim, a interpretação da influência desse parâmetro na evolução temporal da corrente e da tensão nos terminais do semiconductor em estudo. Por fim, o utilizador deve ter liberdade para ampliar ou diminuir a visualização dos gráficos a seu gosto.

As representações gráficas tridimensionais devem apresentar superfícies onde se relacionam a concentração de electrões, com a largura da base do semiconductor e com o tempo de simulação.

O simulador a utilizar deve ser o módulo *SPICE4* [4] do programa de simulação ICAPS [5] e os modelos dos semicondutores devem ser os modelos apresentados e descritos em [6] .

Por fim, a interface a desenvolver deve permitir ao utilizador ter acesso a simulações pré-definidas com variações dos parâmetros de circuito, de carga e de simulação acompanhadas das devidas explicações teóricas.

1.4 - Estrutura do documento

Este documento está organizado em oito capítulos, cujos conteúdos são descritos, de forma sucinta, na presente secção.

O primeiro capítulo é o capítulo introdutório onde consta o enquadramento, os objectivos, os requisitos e a estrutura adoptada para a presente dissertação.

O segundo capítulo consiste no estudo do estado da arte de diversos temas abordados nesta dissertação. Começa-se por descrever um pouco da história da origem dos semicondutores e a sua evolução ao longo do tempo. De seguida, apresentam-se alguns programas educativos, bem como as suas funcionalidades e limitações, terminando-se com a exposição de um programa de simulação de circuitos tradicionalmente usado a nível académico.

No terceiro capítulo - são apresentados os modelos dos semicondutores utilizados, isto é, dos díodos e transístores, sendo que estes estão adequadamente descritos em [6] .

O quarto capítulo tem como objectivo apresentar a ferramenta utilizada na programação da interface de simulação e a ferramenta de simulação utilizada para simular os circuitos pretendidos.

No quinto capítulo apresenta-se a interface de simulação desenvolvida. Nesta secção é dada ênfase à importância de utilizar classes no desenvolvimento do código de suporte à aplicação e à descrição das regras básicas que devem ser tomadas em consideração na implementação de uma interface funcional e adequada ao fim a que se destina [7] [8]. Neste capítulo são ainda apresentados os diagramas UML (*Unified Modeling Language*) de classes, de e de casos de uso [11][9][10][11] , as classes mais relevantes da aplicação, algumas funções

fundamentais para o cumprimento dos objectivos propostos para esta dissertação e por fim o trabalho extra desenvolvido durante a implementação da aplicação, e que visou melhorar o seu funcionamento e as suas potencialidades.

O sexto capítulo tem como primeiro objectivo apresentar e demonstrar o funcionamento da interface desenvolvida, sendo que o segundo passa pela sua validação. A validação é feita comparando o resultado de simulações efectuadas nesta interface com resultados teóricos esperados da teoria dos semicondutores.

O sétimo e último capítulo resume as conclusões retiradas do presente trabalho, sendo também feita a apresentação de algumas sugestões e ideias para trabalhos a desenvolver no futuro.

Capítulo 2

Estado da arte sobre aplicações de simulação de circuitos

2.1 - Introdução

Os materiais semicondutores são aqueles cujas propriedades eléctricas resultam da combinação das que estão inerentes aos metais (condutores) e aos isolantes (não condutores), estando na base de alguns dispositivos electrónicos semicondutores, tais como, díodos e transístores. Estes dispositivos estão presentes numa grande variedade de bens associados ao quotidiano da humanidade, desde um simples rádio ou relógio, até aos mais sofisticados computadores, carros ou aviões, sendo este um facto que, por si só, revela a sua grande importância no dia-a-dia de qualquer sociedade.

A história dos semicondutores é extensa apesar de incidir num período muito curto de tempo (1 século). Embora não haja total concordância, aponta-se meados do século XIX como um marco no avanço desta tecnologia. As primeiras descobertas que revolucionaram e motivaram os cientistas para o estudo e aplicação de semicondutores iniciaram-se no século XVIII com algumas descobertas sobre materiais condutores, isolantes e outros que apresentavam um comportamento intermédio, designados, mais tarde, semicondutores por combinarem características de condutores e isolantes. No entanto, as grandes descobertas sobre os semicondutores e, por conseguinte, sobre os díodos e transístores deram-se apenas nos finais do século XIX e início do século XX, prolongando-se até aos dias de hoje, numa evolução que parece não ter fim à vista.

No sentido de dar seguimento ao tema introduzido no início na presente secção, optou-se por organizar este capítulo da seguinte forma: na secção 2.2, é feito um estudo relativo à origem, evolução e importância dos dispositivos electrónicos constituídos por materiais semicondutores. Na secção 2.3 apresenta-se um programa utilizado em grande escala na simulação de circuitos em meios académicos, o PSIM [12]. Por fim, na secção 2.4, faz-se a exposição de dois programas desenvolvidos recentemente no apoio ao ensino dos

semicondutores, sendo estes, o TESE (*A Tool for Teaching Semiconductor Theory*) [3] e o SEMISOFT (Programa Educativo para o Ensino de Semicondutores) [14].

2.2 - Díodos e transístores - um pouco de História

Com o intuito de facilitar a apresentação cronológica dos factos mais relevantes que viabilizaram o desenvolvimento de componentes electrónicos em torno das características dos materiais semicondutores, esta foi dividida em duas partes (eras). A primeira teve início no século XVIII com as primeiras investigações em torno da condução eléctrica em diferentes materiais, pelos investigadores Stephen Gray (1731), Jean Theophile Desgulliens (1742) e Volta (1782), terminando em 1939 com o início da Segunda Guerra Mundial (SGM). A segunda iniciou-se com o fim da SGM em 1945 e perdura até aos dias de hoje.

Em 1731, Stephen Gray demonstrou que a electricidade estática era transportada por uma gama de materiais interligados entre si. Esta descoberta permitiu a Jean Theophile Desgulliens introduzir, em 1743, o conceito de condutor, e a Alessandro Volta, em 1782, distinguir diferentes materiais e classificá-los, de acordo com a sua condutividade eléctrica, como metais (condutores), semicondutores e isolantes. Alessandro Volta conseguiu ainda inventar, em 1799, a primeira pilha fotovoltaica, capaz de funcionar como uma fonte de corrente constante controlada por tensão. A pilha de Volta teve um grande impacto nas experiências uma vez que melhorou substancialmente as condições disponibilizadas para a sua realização, sendo que, até à data, estas dependiam apenas da energia electrostática [20] [26].

Depois de um período superior a trinta anos sem se registar qualquer descoberta, Michael Faraday, em 1833, descobriu um comportamento especial num composto de sulfato de prata. Faraday verificou que este composto tinha um coeficiente negativo de resistência com a temperatura, ou seja, a resistência à passagem da corrente eléctrica do composto diminuía quando a temperatura aumentava (o que se veio a verificar, mais tarde, como sendo um comportamento típico dos semicondutores). Outra importante contribuição para o estudo da física dos semicondutores foi desenvolvida por Alexander Edmond Becquerel, em 1839, com a descoberta do efeito foto voltaico em eléctrodos de platina cobertos por cloreto de prata. Este processo foi detectado ao estabelecer-se o contacto entre um semicondutor de cloreto de prata e um condutor de prata. De referir ainda o facto destas descobertas terem aberto definitivamente a porta para o estudo dos semicondutores [26].

No ano de 1873, Frederick Guthrie descobriu que um eléctrodo com cargas positivas podia ser completamente descarregado quando colocado perto de um pedaço de metal quente, sem que houvesse contacto físico entre eles. Além disso, constatou que se aplicasse uma tensão negativa ao mesmo eléctrodo, este efeito não se verificava, sendo que este facto era indicativo de que a corrente apenas fluía num sentido. Este efeito ficou conhecido por efeito

termiónico, estando na base dos díodos termiónicos tendo sido, inclusivamente, redescoberto acidentalmente por Thomas Edison, em 1880.

A lâmpada de Edison era composta por um filamento muito fino em carbono inserido numa redoma de vidro fechada em vácuo. O aquecimento do filamento de carbono era responsável pela emissão de luz. Edison verificou que fragmentos de carbono eram libertados do filamento incandescente e depositavam-se na redoma de vidro, como se pode verificar na Figura 2-1.

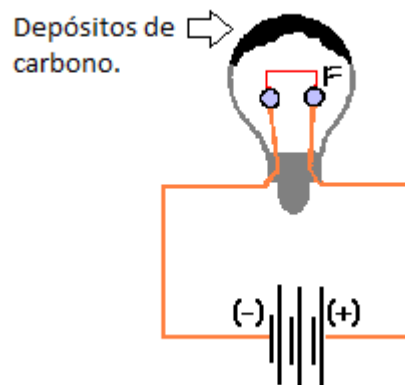


Figura 2-1- Esquema da lâmpada de Edison [30].

Com o objectivo de reduzir este fenómeno de deposição de fragmentos de carbono, Edison decidiu introduzir dentro da redoma de vidro uma placa de metal ligada ao terminal positivo da fonte de tensão, como se pode verificar na Figura 2-2. Edison introduziu ainda um galvanómetro¹ para verificar se alguma corrente eléctrica fluía da bateria para a placa de metal ou vice-versa.

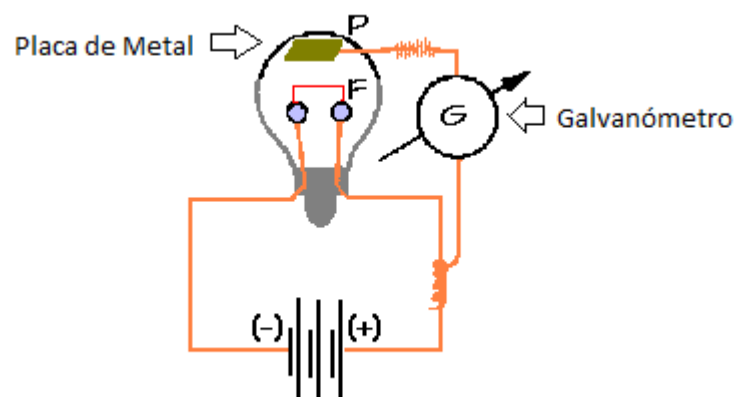


Figura 2-2 - Esquema da Lâmpada de Edison com a introdução da placa de metal [30].

¹ O galvanómetro é um aparelho de medida da corrente eléctrica, inventado por Faraday e é composto por uma bobine móvel (fio muito fino) montada num eixo móvel e instalada entre dois pólos de um íman. Quando circula uma corrente eléctrica pela bobine, forma-se um campo magnético que interage com o campo magnético criado pelo íman e a bobine gira, movendo um ponteiro sobre uma escala graduada. O movimento do ponteiro é proporcional à corrente eléctrica que percorre a bobine e é indicado na escala graduada.

Com esta experiência, Edison constatou que a corrente eléctrica podia ser extraída do filamento incandescente e transportada, através do vácuo, até à placa de metal. Contudo, este efeito só ocorria quando era aplicada uma tensão positiva aos terminais da placa de metal. Estes fenómenos foram os primeiros indícios que permitiram afirmar que o comportamento do díodo tinha sido descoberto, embora na altura não se percebesse qual poderia ser a sua utilidade [15][16][21].

Estas descobertas pioneiras foram a base para o verdadeiro desenvolvimento de dispositivos electrónicos, que se iniciou em 1874 quando o alemão Karl Ferdinand Braun, durante os seus primeiros anos como professor, encetou os primeiros estudos sobre as propriedades eléctricas dos compostos de sulfato de chumbo e sulfato de cobre. Do seu trabalho de investigação resultou o desenvolvimento de um rectificador construído com um cristal de sulfato de chumbo, sendo que após a realização das primeiras experiências com este mesmo rectificador foi possível comprovar que a corrente eléctrica fluía de maneira diferente de acordo com o sentido da diferença de potencial aplicada na alimentação deste dispositivo. Com esta descoberta, Braun contrariava por completo a lei de Ohm, no sentido em que esta afirmava que a diferença de potencial entre dois pontos de um condutor é proporcional à corrente que o percorre, enquanto que a descoberta de Braun mostrava que a corrente eléctrica fluía de maneira diferente conforme o sentido da diferença de potencial aplicado aos terminais do dispositivo. Com efeito, o jovem professor alemão implementou um rectificador que ficou conhecido como *Cat's Whisker Rectifier*, rectificador 'bigodes gato', apresentado na Figura 2-3. Fazendo um enquadramento entre o *Cat's Whisker Rectifier* e a história contemporânea da humanidade, refira-se que este desempenhou um papel muito importante no desenvolvimento de radares utilizados na SGM.



Figura 2-3- Rectificador de “Bigodes de Gato”, por Braun [15].

Outros fenómenos de rectificação foram observados, ainda em 1874, por Arthur Schuster e por Charles Edger Fritts, em 1876. O facto de diferentes investigadores, residentes em locais geograficamente distantes, registarem a ocorrência destes fenómenos indiciava que as descobertas a eles associadas não eram um mero acaso. Este foi um dos factores que impulsionou o crescimento do número de actividades de investigação, sendo que se foram

obtendo ao longo do tempo resultados e descobertas importantes, como por exemplo a primeira pilha solar de selénio, criada por Charles Edger Fritts, em 1878.

O efeito de Hall (Figura 2-4), estabelecido pelo físico que lhe deu o nome, Edwin Herbert Hall, em 1879, aparece como um marco importante na história dos semicondutores. Hall aproveitou um estudo deixado pelo seu professor H. Rowland que visava compreender o efeito da aplicação de um campo magnético sobre uma corrente eléctrica. Rowland acreditava que a aplicação de um campo magnético perpendicular à direcção na qual se faz o fluxo da corrente, provocaria um desvio nas cargas carregadas, criando, assim, uma diferença de potencial. Todavia, na altura da sua descoberta, o efeito de Hall não era interpretado como nos dias de hoje, uma vez que não se sabia qual a natureza das cargas (eléctricas) em causa.

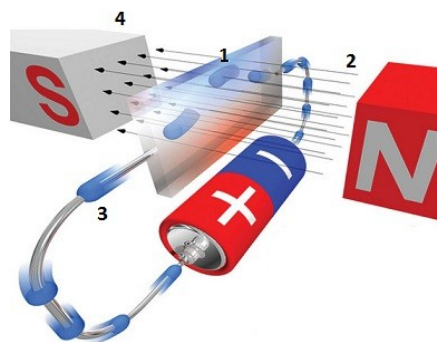


Figura 2-4- Efeito de Hall. 1- Carga sofre um desvio provocado pelo campo magnético, 2-Campo magnético, 3- Cargas, 4-Ímans.

No ano de 1897, Joseph Thompson anunciou a descoberta do *electron* e explicou que a corrente eléctrica viajava através do espaço, num determinado sentido e num tubo de vácuo, corroborando o efeito de Edison. Posteriormente, em 1904, John Ambrose Fleming enquanto estudava o efeito de Edison, patenteou um dispositivo que é conhecido como sendo o primeiro dispositivo electrónico: A Válvula de Fleming - Figura 2-5. De referir que aparecem também as designações de Díodo de Vácuo ou Válvula Termiónica para este mesmo dispositivo.



Figura 2-5- Válvula de Fleming [22].

Fleming usou o efeito de Edison para rectificar um sinal eléctrico de fraca intensidade, emitido por rádio frequência. Tal como Edison, Fleming verificou que a rectificação dos sinais de rádio frequência só acontecia num sentido, através da variação do valor da corrente apresentado no galvanómetro. O rectificador de Fleming foi bastante utilizado na transmissão rádio, substituindo os rectificadores descobertos por Braun (Cat's Whisker Rectifier). Porém, o díodo de vácuo apresentava algumas vantagens e desvantagens quando comparado com o díodo de bigodes de gato. No díodo de bigodes de gato, a necessidade de escolher o local exacto para fazer as ligações eléctricas implicava que qualquer vibração mais forte as pudesse quebrar, tornando-o mais frágil e menos confiável. Assim, o díodo de vácuo começou a ganhar vantagem, passando a ser mais utilizado em detrimento, por exemplo, do Cat's Whisker Rectifier. Todavia, o díodo de vácuo necessitava de uma fonte de tensão externa capaz de aquecer o seu filamento à temperatura necessária, sendo que este era um dos factores que representava uma desvantagem face aos díodos bigode de gato. No que toca à utilização de díodos a frequências elevadas, os díodos de bigode de gato levavam também vantagem quando comparados com os díodos de vácuo [22][15][16][18].

Na sequência dos anteriores, outro avanço científico relevante foi a invenção de Lee de Forrest, em 1906, que consistiu na criação de um novo tipo de díodo - o tríodo. Este era composto por mais um eléctrodo, sob a forma de uma grade de fios, entre o filamento (cátodo) e a placa de metal (ânodo). A introdução deste terceiro eléctrodo permitiu que a corrente do ânodo fosse controlada independentemente através da aplicação de uma pequena tensão entre o cátodo e o eléctrodo extra, permitindo que este díodo de vácuo tivesse a capacidade de amplificar os sinais eléctricos. Na Figura 2-6 apresentam-se, de forma simplificada, o efeito de Edison, a válvula de Fleming e o tríodo de Lee de Forrest.

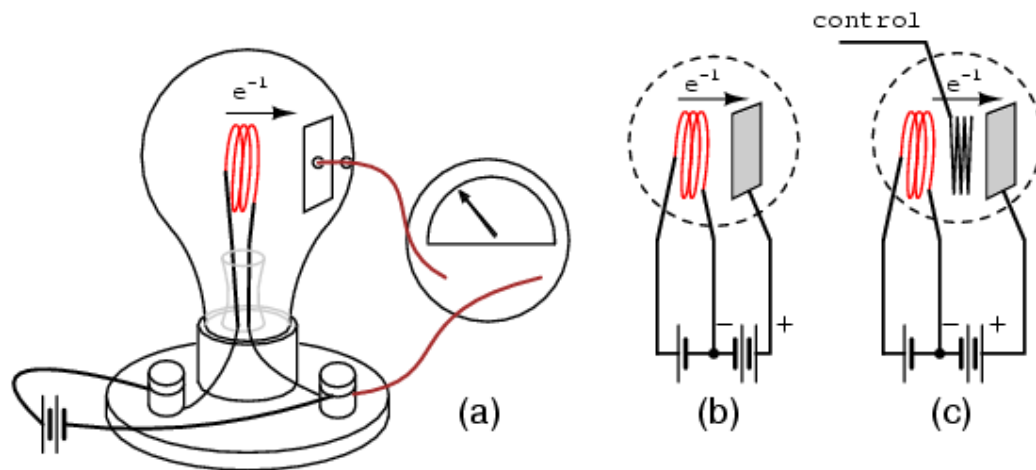


Figura 2-6- a)-Efeito de Edison, b)- Válvula de Fleming, c)- Tríodo de Lee de Forrest [31].

O início do século XX revelou-se fundamental para o desenvolvimento da electrónica, tanto a nível de sinal como de potência. Depois de alguns anos em que se multiplicaram o número de experiências em torno destes novos componentes, havia necessidade de desenvolver as teorias que explicassem os fenómenos descobertos e sustentassem o funcionamento destes mesmos componentes.

A teoria desenvolvida sobre a mecânica quântica durante a década de vinte pelos físicos Bohr, Broglie, Heisenber e Schrodinger, foi, então, a chave encontrada para o estudo da electrónica. Com esta teoria foi possível criar os conceitos de bandas de energia, bandas proibidas, banda de valência, portadores, etc. Em conjunto com a teoria quântica dos sólidos, liderada por Peirls, Wilson, Mott e Franck, entre outros, foi possível compreender a condutividade electrónica em metais e permitir desenhar uma imagem da estrutura electrónica dos mesmos [17].

Em paralelo com o desenvolvimento das teorias acima apresentadas, foi proposto por Lilienfeld em 1926, o primeiro conceito de um transístor de efeito de campo, conhecido ainda hoje por FET (*Field Effect Transistor*). Lilienfeld acreditava que a aplicação de uma tensão eléctrica num material de fraca condutividade iria mudar a sua capacidade de condução, alcançando-se, assim, a amplificação. Embora ele tenha patenteado este conceito, ninguém, nem ele próprio, conseguiram na altura tirar qualquer proveito [17].

No final da década de trinta, mais precisamente no ano de 1936, Mervin Kelly decidiu criar um grupo de investigadores nos laboratórios da Bell Labs. O grupo era formado por investigadores como Bill Shockley, Russel Ohl, Jack Scaff, entre outros. O líder deste grupo, Mervin Kelly, tinha o pressentimento de que os rectificadores de vácuo não eram a última descoberta da electrónica. Em 1940, Russel Ohl, um dos investigadores do grupo da Bell Labs, na tentativa de compreender os cristais de silício, aprendeu que, dependendo da forma como estes são preparados (quantidade e qualidade das impurezas), podem-se obter rectificadores

positivos (p) ou rectificadores negativos (n), de acordo com o sentido em que é feita a rectificação, ou seja, de acordo com a teoria da mecânica quântica, Ohl percebeu que havia cristais de silício cuja banda de valência era composta por cargas positivas (tipo p) e outros por cargas negativas (tipo n). Esta foi a última descoberta antes do início da SGM.

Com o início da SGM, o estudo sobre a electrónica estagnou e focalizou-se apenas em vertentes relacionadas com a guerra, como por exemplo, em equipamentos como o radar. O grupo de Kelly da Bell Labs dividiu-se e a primeira era de descobertas em torno dos semicondutores tinha chegado ao fim. Na Figura 2-7 representa-se a evolução histórica registada ao longo desta mesma era.



Figura 2-7- Diagrama temporal da evolução dos dispositivos electrónicos constituídos por semicondutores antes da Segunda Guerra Mundial

Com o fim da SGM, em 1946, Kelly reuniu outra equipa de investigadores liderados por Bill Shockley e Morgan Stanley. A equipa era ainda constituída por Walter Brattain, John Bardeen, John Pearson, Bert Moore e Robert Gibney. A primeira decisão importante tomada por esta equipa foi considerar que o silício e o germânio eram os materiais semicondutores mais simples, optando-se por focalizar o seu estudo apenas nestes. Bardeen compreendeu os efeitos de campo e mostrou, através de cálculos, que a aplicação de uma tensão num material semiconductor implicava uma mudança da sua condutividade.

Foi então que, em 16 de Dezembro de 1947, Bardeen e Brattain apresentaram o primeiro transístor. Ironicamente, o objectivo inicial era fabricar um FET, mas acabaram por descobrir que a corrente de saída era maior do que a corrente de entrada, ou seja, tinham descoberto o fenómeno de amplificação.

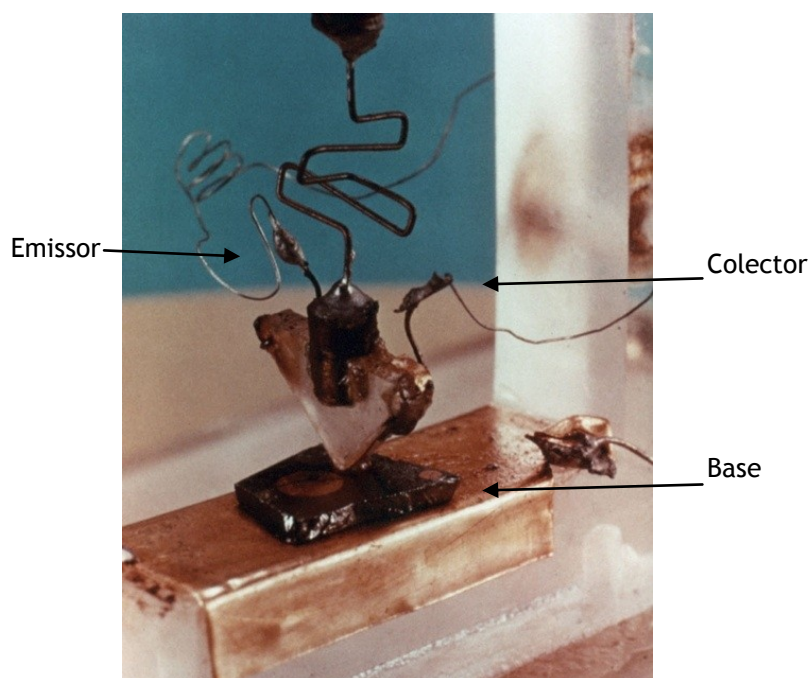


Figura 2-8 - Transístor desenvolvido por John Bardeen e Walter Brattain [32].

Pela análise da Figura 2-8 pode-se verificar que este transístor era um dispositivo complexo. O eléctrodo do fundo, designado por base, era um cristal de germânio, estando os outros condutores aplicados por cima deste. A ponta de contacto era composta por um condutor, seguida de uma camada de cera isolante e, por último, de uma camada de material condutor. Assim, o condutor interior era o colector, enquanto que o condutor exterior era o emissor. Esta foi a origem dos termos que ainda hoje surgem de forma mais frequente aquando da utilização e estudo dos transístores. Utilizando o transístor desenvolvido, esta equipa de investigadores realizou algumas experiências. Na primeira experiência, ligaram o condutor externo a um eléctrodo, tendo-se verificado que a corrente que fluía do emissor

para o colector era controlada pela corrente da base. Posteriormente, Brattain percebeu que se utilizasse duas tiras de metal muito próximas na superfície do germânio, ele teria a mesma configuração, mas com uma complexidade muito inferior. A Figura 2-9 apresenta um esquema do primeiro transístor desenvolvido por Bardeen e Brattain.

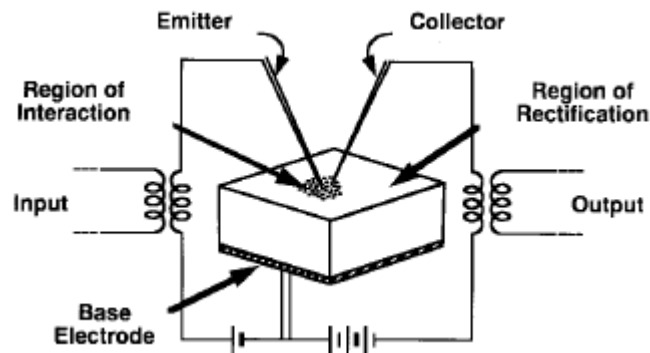


Figura 2-9 - Esquema do primeiro transístor [17].

Depois de comprovados estes resultados, decidiram estudar e perceber o efeito de amplificação de sinais eléctricos. Por um lado, Bardeen pensava que a amplificação era causada pela mudança de condutividade da camada superficial, sendo que, por outro lado, Shockley estava certo que o cristal de germânio era o grande responsável por esta operação. Assim, ao longo dos meses que se seguiram, Shockley começou escrever uma teoria que ficou conhecida como teoria do transístor de junção bipolar [17]. Esta teoria dependia claramente da introdução de portadores minoritários nos materiais semicondutores que estão na base dos transístores. Os portadores minoritários são portadores que estão em menor número nos materiais semicondutores, ou seja, num material semicondutor do tipo N, as cargas negativas (electrões) estão em maioria (portadores maioritários), enquanto que as cargas positivas (lacunas) estão em minoria (portadores minoritários). Num material semicondutor do tipo P, os portadores maioritários são as lacunas e os minoritários, os electrões.

Posteriormente, John Shive validou a teoria de Shockley. Shive, por sua vez, pegou num pedaço muito fino de germânio e colocou o emissor e o colector em lados opostos do cristal, eliminando os caminhos de superfície. Com esta experiência, Shive verificou que o que estava a acontecer realmente era uma injeção de portadores minoritários, corroborando, desta forma, a teoria do transístor bipolar de junção (TBJ), desenvolvida por Shockley. A Figura 2-10 apresenta o esquema do primeiro (TBJ).

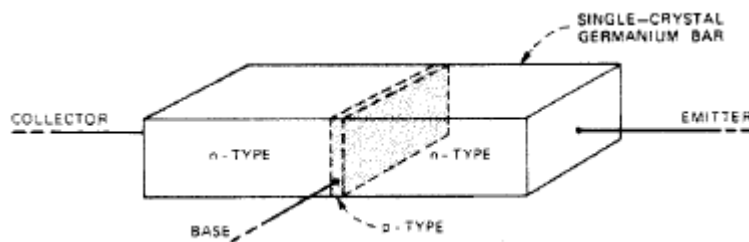


Figura 2-10- Esquema do primeiro transístor bipolar de junção [17].

Em Abril de 1950, Brattain, Shockley, Teal e Sparks construíram o primeiro TBJ (com polarização) NPN (*Negative, Positive, Negative*), baseados na teoria desenvolvida por Shockley.

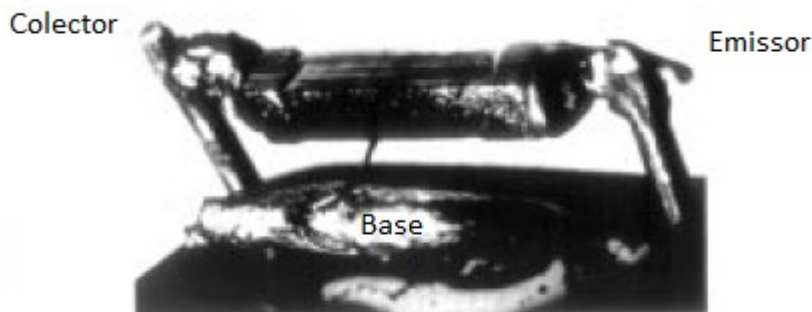


Figura 2-11- Transístor de junção bipolar [17] .

Apesar da descoberta do transístor bipolar se ter tornado uma realidade, as pesquisas para a obtenção de um transístor de efeito de campo mantiveram-se, sendo que em 1952, Ian Ross e George Dacey implementaram um transístor de efeito de campo unipolar. Com este transístor contornou-se a problemática em torno dos estados de superfície que até à data eram uma incógnita. Este dispositivo foi o precursor do FET que hoje se utiliza em larga escala.

Um facto que contribuiu bastante para o desenvolvimento dos semicondutores foi o facto de a Bell Labs ter vendido as licenças da produção de transístores a outras empresas. Assim, empresas como a Texas Instruments e a Sony compraram licenças para aprender a usar a tecnologia de fabrico de transístores. Como consequência desta proliferação no fabrico dos transístores, também o mercado de consumo para os mesmos se começou a desenvolver. Neste campo, destaca-se a título de exemplo a Sony que, na altura, foi a primeira empresa a comercializar um rádio totalmente transistorizado. Outras empresas foram sendo criadas durante esta altura, destacando-se, entre elas, a Intel e a Fairchild.

Controlados os processos de fabrico de transístores, nasceu a ideia de criar um circuito que fosse capaz de integrar vários componentes electrónicos. Este conceito foi proposto por Jack Kilby, investigador da Texas Instruments (TI), em 1958. Kilby demonstrou a sua ideia

através de um circuito fabricado em cima de um único bloco de silício, contendo um transístor (do tipo mesa), um condensador e uma resistência.

Os transístores do tipo mesa tinham sido desenvolvidos na TI, em 1957, sendo os primeiros a possuir duas bases difusas e emissores difusos. Os três elementos constituintes deste circuito eram, no entanto, interligados por fios soldados nos seus cantos. O primeiro circuito integrado (CI) era semelhante ao da Figura 2-12, embora tenha pouco a ver com os CIs que hoje existem.

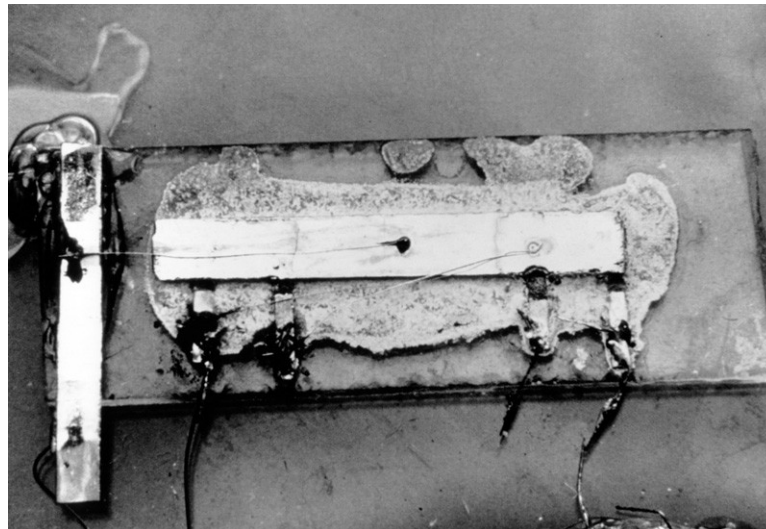


Figura 2-12 - Primeiro circuito integrado desenvolvido por Jack Kilby's [17].

Paralelamente, um grupo de investigadores da Fairchild trabalhava no desenvolvimento de um processo superior para o fabrico de transístores, o processo planar. Este processo revolucionou a fabrico de CIs e foi fundamental para o desenvolvimento do mercado dos mesmos.

No final da década de cinquenta, outras descobertas se vieram a revelar importantes. O estudo e desenvolvimento de processos de oxidação do silício possibilitaram o desenvolvimento do transístor de efeito de campo, com porta isolada, ou seja, o MOSFET (*Metal Oxide Semiconductor Field Effect Transistor*), ou simplesmente MOS (*Metal Oxide Semiconductor*). A interface entre o dióxido de silício (SiO_2) e o silício (Si) era de boa qualidade com baixa densidade de estados de superfícies (que apareciam com muita frequência nos FET's). Todavia, os dispositivos MOS apresentavam uma fraca estabilidade, facto que originou um atraso superior a 10 anos na sua utilização em larga escala. Esta estabilidade estava directamente relacionada com o pouco controlo das impurezas, mais especificamente as de sódio (Na) existentes no material semiconductor utilizado. Assim, os problemas encontrados deviam-se aos materiais e não ao processo de produção, pelo que era urgente controlar e evitar os efeitos devastadores que o sódio provocava na interface SiO_2/Si . Na área do desenvolvimento, algumas descobertas permitiram criar, em 1961, um circuito

integrado que já se aproximava mais dos que hoje são fabricados, como se pode verificar na Figura 2-13.

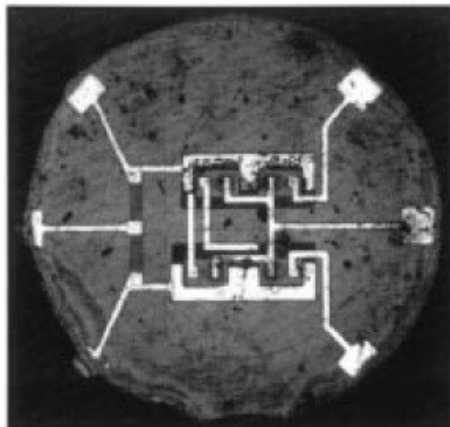


Figura 2-13- Primeiro CI desenvolvido pela Fairchild através de um processo planar [17].

A combinação de transístores MOS de canal n e canal p num mesmo substrato levou F. Wanlass a propor a tecnologia CMOS em 1963.

Vários dispositivos electrónicos foram descobertos devido aos requisitos específicos relativos a diversas e distintas aplicações. Porém, deve-se salientar que as tecnologias MOSFET e TBJ dominaram o período compreendido entre 1974 a 1986, registando-se um decréscimo das tecnologias TBJ e um aumento da tecnologia MOSFET, em particular da tecnologia CMOS, na viragem do século 20 para o século 21.

A evolução dos circuitos integrados nos dias de hoje não passa apenas por desenvolver novos componentes. A redução contínua do tamanho dos CI e dos seus constituintes, o aumento do número de componentes por CI e a evolução do empacotamento dos CI são factores relevantes e que marcam o estudo actual dos dispositivos constituídos por materiais semicondutores. A combinação destes três factores levou a um aumento notável do número de componentes que é possível colocar em cada CI, sendo que a tendência de crescimento é a duplicação em cada dezoito meses. De referir que Gordon Moore, enquanto presidente da Intel, fez esta previsão, em 1965, tendo dela resultado a Lei de Moore (Figura 2-14) [40].

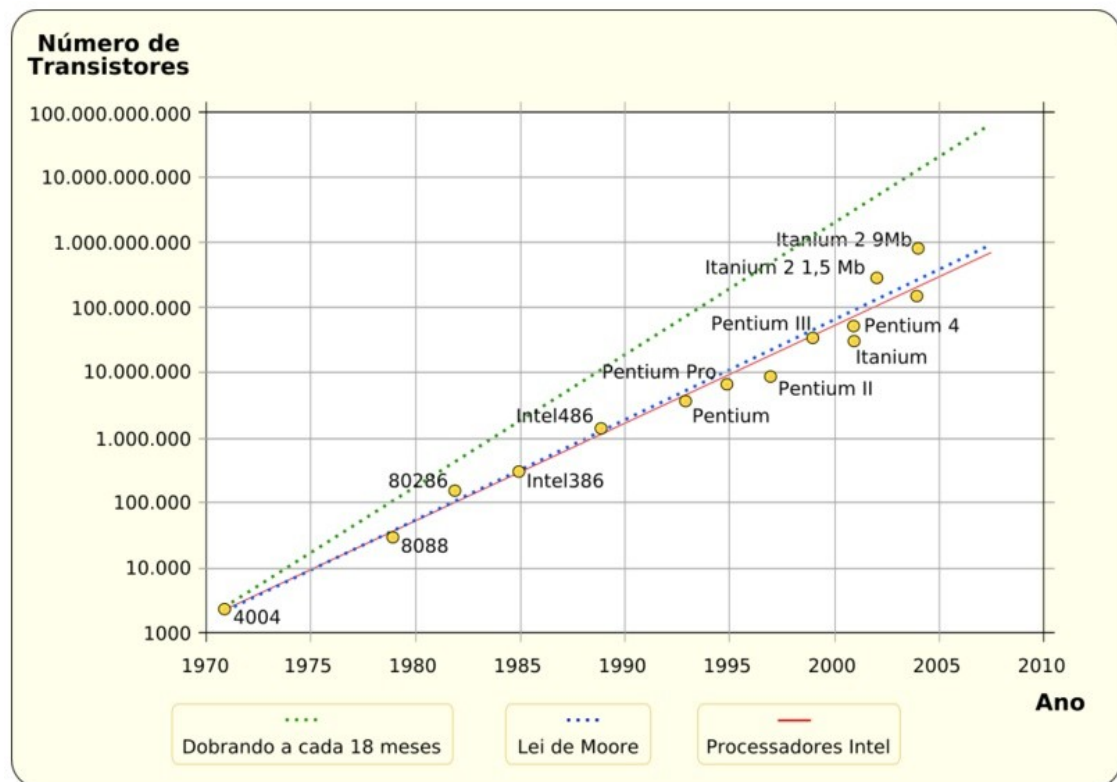


Figura 2-14- Lei de Moore [33].

A evolução dos CIs passa não só pelo aumento dos requisitos a nível de integração dos componentes electrónicos, tal como já foi referido supra, mas também pelo objectivo de se garantirem maiores velocidades de operação dos mesmos, para além de menores consumos de potência energética.

Tendo em conta o tamanho das moléculas (da ordem dos nanómetros) e os objectivos dos CIs, alguns investigadores lembraram-se de estudar a viabilidade de criar dispositivos electrónicos de tamanho molecular. No seguimento deste trabalho de investigação, uma equipa de investigadores da Universidade do Estado de Arizona, nos Estados Unidos, construiu o primeiro díodo molecular, em Outubro de 2009 [36]. A Figura 2-15 apresenta o esquema do díodo molecular.

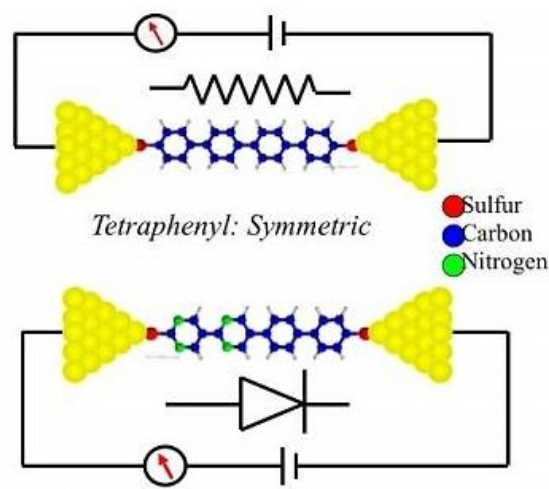


Figura 2-15 - Esquema do díodo molecular desenvolvido na Universidade do Estado do Arizona [36].

Posteriormente, em 23 de Dezembro de 2009, alguns investigadores da Universidade de Yale, nos Estados Unidos apresentaram o primeiro transístor molecular, tendo sido apenas produzido um número suficiente para a realização de testes em laboratório[36].

Embora estes trabalhos já decorressem há mais de duas décadas, só com o avanço da tecnologia é que foi possível fazer a medição de algumas grandezas eléctricas, como é o da corrente, e visualizar o que está acontecer ao nível molecular (Figura 2-16). Este transístor é composto por uma molécula de benzeno ligada a contactos de ouro. Assim, é possível manipular os diferentes estados de energia da molécula variando a tensão aplicada nos eléctrodos de ouro. Ao controlar os estados de energia, consegue-se controlar a intensidade de corrente que passa através da molécula. Este comportamento é em tudo semelhante ao de um transístor electrónico de silício.

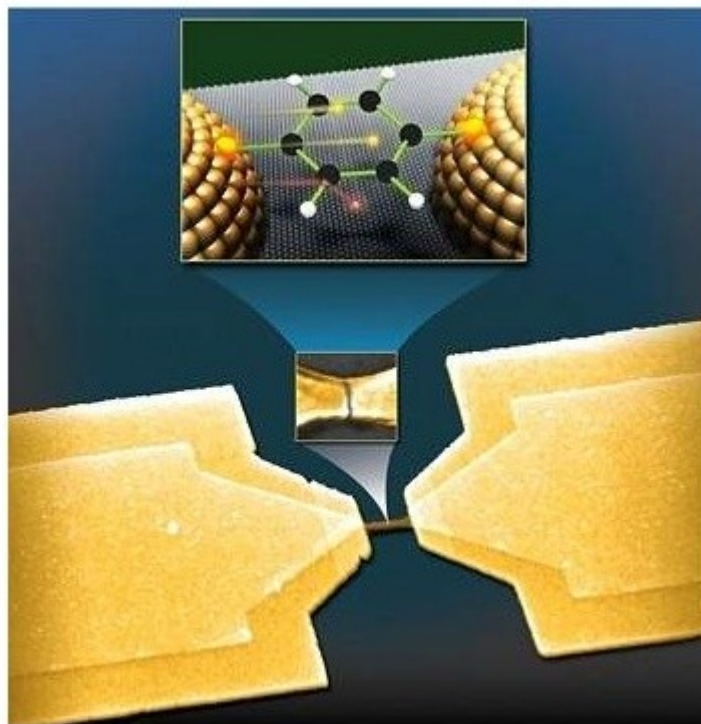


Figura 2-16- Visualização da variação da estrutura do benzeno quando se aplica uma tensão aos eléctrodos de ouro [37].

No sentido de encontrar materiais que, de alguma forma, apresentassem melhores comportamentos eléctricos do que o silício, inúmeras experiências e investigações foram realizadas. Neste sentido, surgiu um novo elemento, o grafeno. O seu estudo começou em 2004, altura em que os investigadores implementaram o primeiro transístor deste material. No entanto, esta primeira tentativa foi um fracasso, uma vez que o transístor não podia ser desligado [34] [39] . Refira-se que o grafeno é uma estrutura plana composta unicamente por átomos de carbono. Estes átomos estão dispostos numa estrutura hexagonal gerando uma folha plana (Figura 2-17).

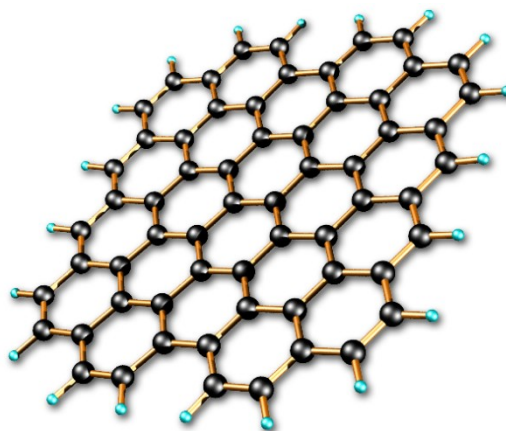


Figura 2-17- Estrutura do grafeno [38] .

As experiências realizadas têm mostrado que o grafeno tem uma mobilidade de electrões superior à do silício. Além disso, um dos pontos também de elevado interesse é o facto deste material quando reduzido a apenas alguns nano metros de largura conseguir manter-se estável, ao invés do que acontece com o silício.

A empresa Fujitsu apresentou recentemente (Janeiro de 2010) o primeiro protótipo de um CI construído com grafeno [39]. Assim, espera-se que o grafeno seja o substituto do silício, mantendo de pé a lei de Moore, ou seja, a possibilidade de se integrar cada vez mais componentes electrónicos por CI.

A Figura 2-18 apresenta o diagrama temporal da evolução dos dispositivos compostos por semicondutores no período que se seguiu à Segunda Guerra Mundial.

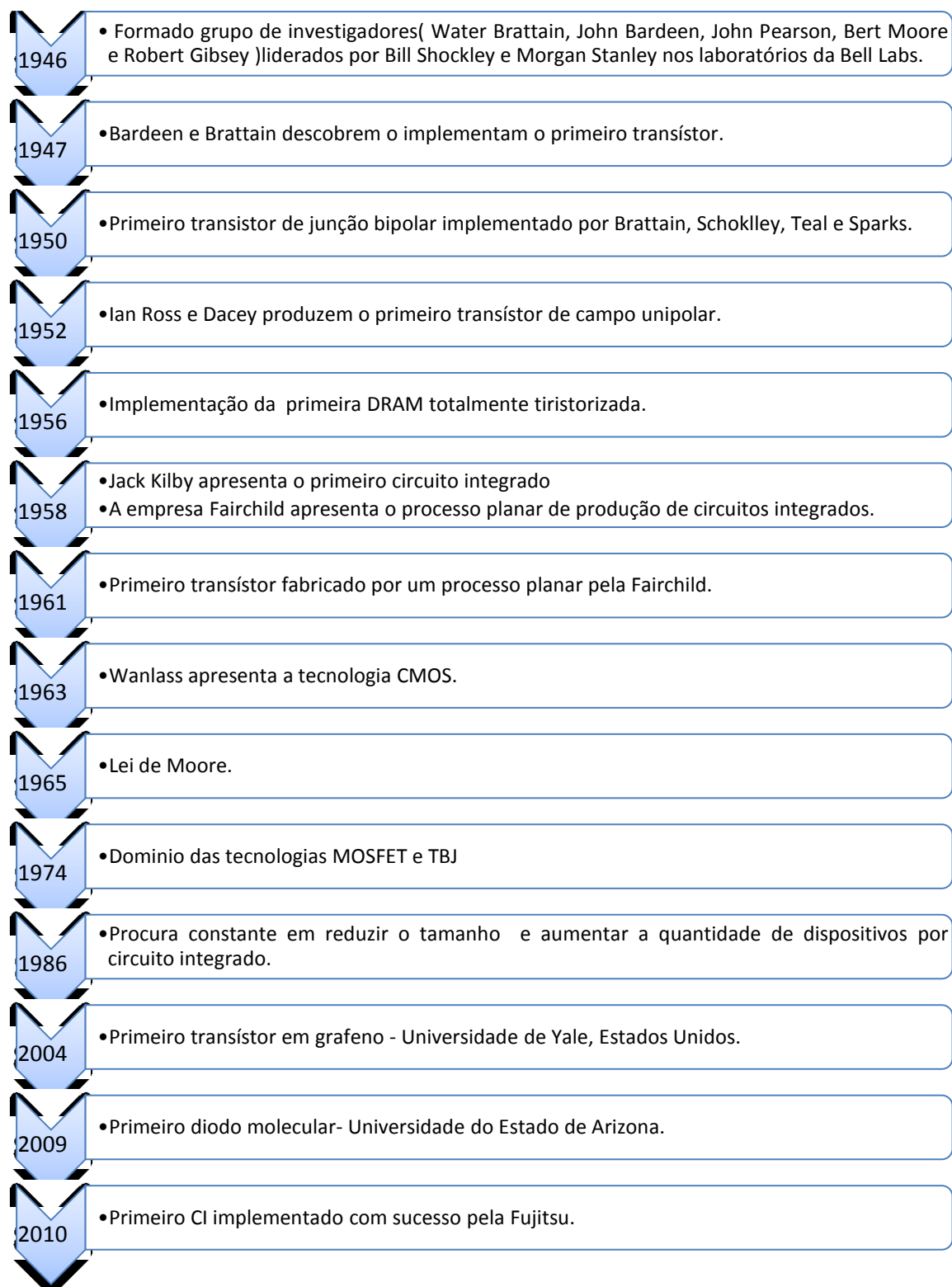


Figura 2-18- Diagrama temporal da evolução dos dispositivos electrónicos constituídos por materiais semicondutores depois da Segunda Guerra Mundial.

2.3 - Aplicações educacionais

Nesta secção serão apresentados alguns programas educativos publicados na literatura desenvolvidos para apoiar o ensino de semicondutores, nomeadamente o TESE [3] e o SEMISOFT [14].

2.3.1 - TESE - Teaching of Semiconductors

O TESE [3] é um programa que apresenta como principais funcionalidades o facto de permitir explorar os aspectos fundamentais sobre a teoria dos semicondutores e da análise de circuitos de potência. Com esta ferramenta é possível analisar a influência dos parâmetros dos dispositivos semicondutores e dos parâmetros do circuito de potência em comportamento dinâmico, sendo também possível analisar as formas de onda típicas relativas às principais grandezas eléctricas inerentes a estes mesmos dispositivos e circuitos.

O programa desenvolvido utiliza um simulador de circuitos (IsSpice) para simular um circuito com valores definidos pelo utilizador. Os modelos dos semicondutores utilizados para as simulações estão descritos em [6]. Os resultados da simulação são apresentados através de gráficos, de tensão e corrente, de forma a tornar clara a influência de determinado parâmetro do dispositivo semiconductor no seu próprio comportamento dinâmico. Assim, e segundo o autor do TESE, com esta abordagem não é necessário que o utilizador tenha qualquer conhecimento prévio sobre o simulador utilizado, podendo focalizar a sua atenção apenas na influência de determinado parâmetro no funcionamento global do circuito [3].

A ferramenta foi desenvolvida com o Microsoft Visual Studio 2005 (Visual Basic) e foi organizada em dois níveis de interacção: o módulo cliente e o módulo administrador. O primeiro é o módulo principal e permite ao utilizador seleccionar alguns circuitos pré-definidos para, posteriormente, os estudar e testar. Por sua vez, o segundo permite ao utilizador adicionar mais circuitos, aumentando, assim, o lote dos que aparecem disponíveis, por pré-definição, em utilizações futuras. De referir ainda que os circuitos estão organizados por dispositivo, isto é, quando o utilizador selecciona um, seja ele diodo ou IGBT (aparecem todos os circuitos disponíveis relacionados com o escolhido. O circuito apresentado não é uma imagem estática, podendo-se manipular todos os seus componentes. Desta forma, o utilizador pode escolher individualmente cada componente e definir mais do que um valor para o mesmo, como se pode ver na Figura 2-19.

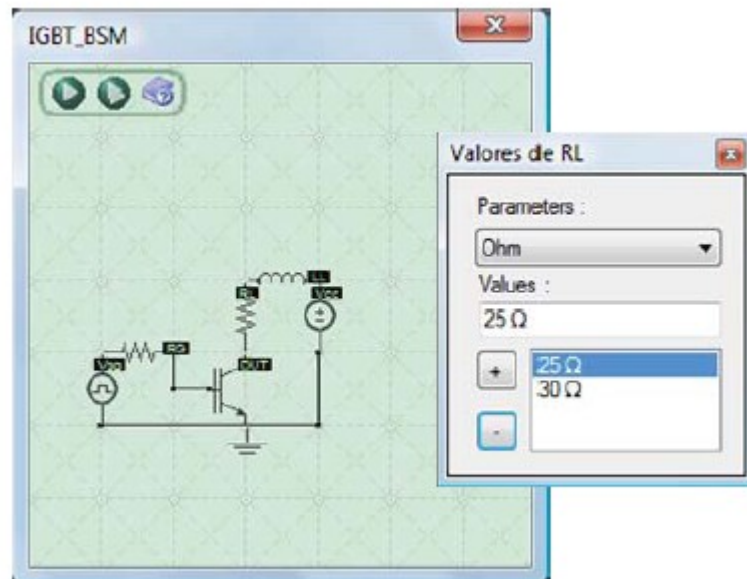


Figura 2-19 - Possibilidade de definir valores para o componente seleccionado [3].

Depois de definir os valores que pretende para cada componente, todas as combinações possíveis são construídas e as simulações são executadas em conformidade (“Supondo que dois valores são fixados para a carga resistiva e dois valores para o tempo médio de vida de um dispositivo, então um total de quatro simulações vão ser feitas, correspondendo a um total de quatro combinações possíveis” [3]).

2.3.2 - SEMISOFT - Semiconductor Programa

O SEMISOFT (Figura 2-20), tal como a aplicação TESE, permite explorar os aspectos fundamentais da teoria dos semicondutores [14]. Este programa utiliza o mesmo simulador de circuitos (IsSpice) para simular circuitos pré-definidos e os mesmos modelos de simulação, descritos em [6]. Os resultados das simulações são disponibilizados através de gráficos 2D (tensão e corrente) e em gráficos 3D (tempo de simulação, concentração e largura da base do semiconductor). Estes gráficos são posteriormente guardados num histórico de simulações. O SEMISOFT tem ainda a particularidade de possibilitar a introdução de novos circuitos. Para tal é necessário que um ficheiro de simulação (ficheiro utilizado pelo IsSpice) seja devidamente manipulado e adaptado por um utilizador avançado.

A grande mais-valia do SEMISOFT face ao TESE é, de facto, a possibilidade de gerar gráficos 3D, ou seja, é gerada uma superfície que relaciona os valores da concentração de electrões/lacunas com a largura da zona de armazenamento de carga (n^-) em função do tempo, permitindo assim uma interpretação mais física dos fenómenos observados pelos gráficos 2D, em particular, o da corrente e tensões. No entanto, esta representação 3D

apresenta algumas limitações importantes que serão mencionadas na descrição mais detalhada da aplicação SEMISOFT que se faz nos parágrafos seguintes.

O programa é composto por vários menus (Início, Simulação, Circuito Genérico, Configurar). A janela de visualização apresentada corresponde ao menu “Início”.

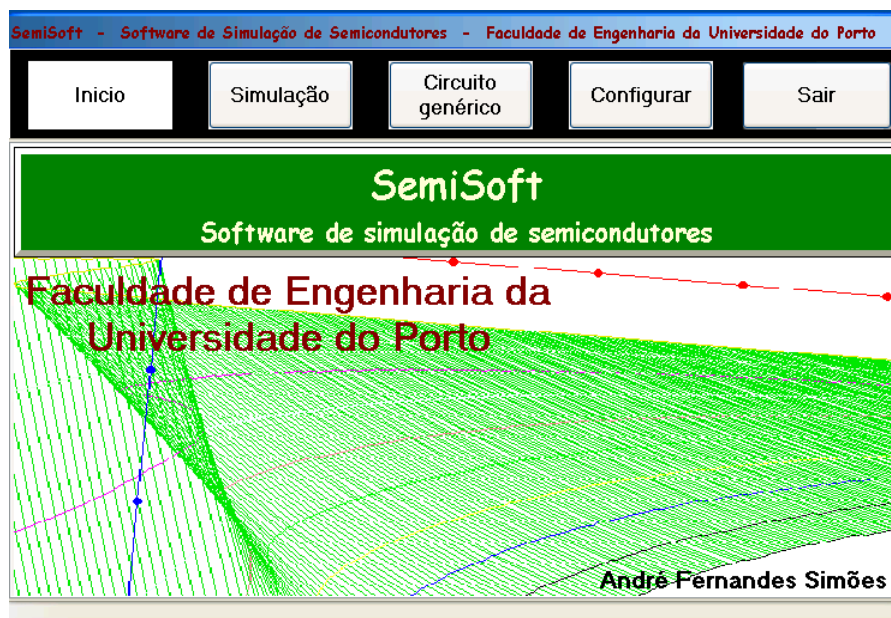


Figura 2-20- Menu Início do programa SEMISOFT.

Ao nível do menu Simulação, é possível simular três tipos de circuitos, um com um diodo, outro com um IGBT e outro com um BJT, como se pode comprovar pela análise da Figura 2-21.

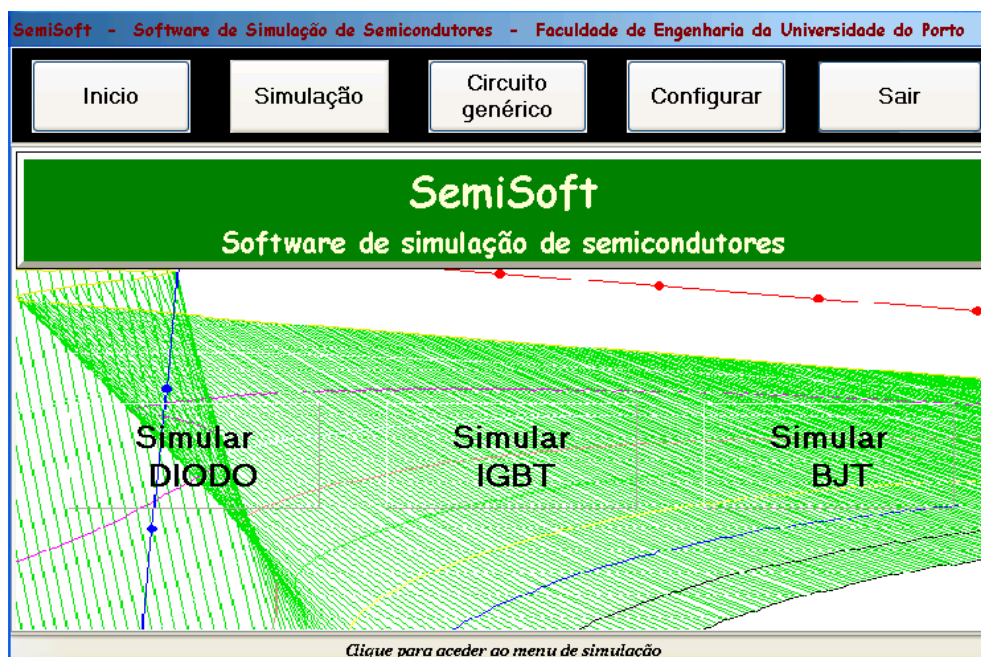


Figura 2-21 - Menu de simulação, SEMISOFT.

Ao carregar no botão “Simular DIODO”, abrir-se-á uma página com um circuito contendo um diodo Figura 2-22. Como se pode verificar estes menus apresentam uma estrutura similar à do Windows XP com o objectivo de facilitar e tornar a navegação no programa mais intuitiva para o utilizador.

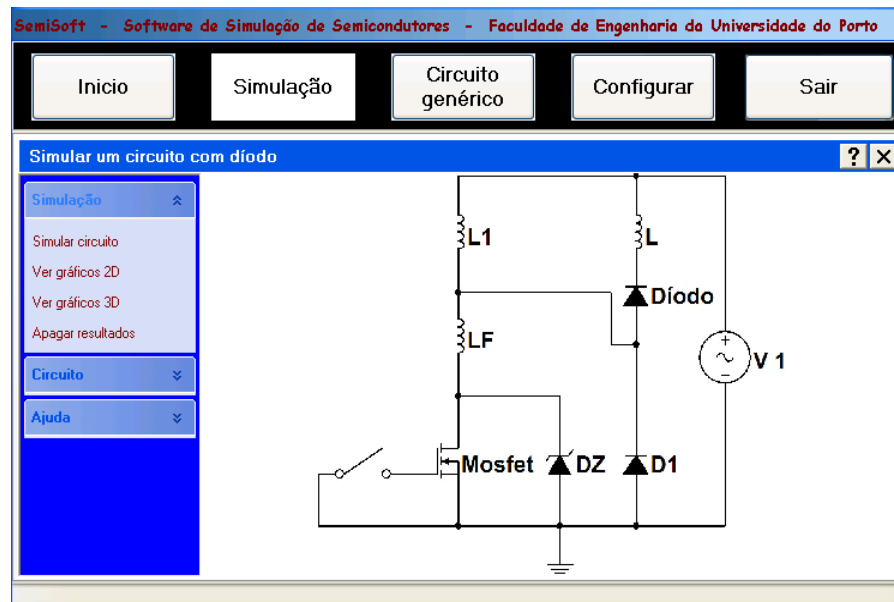


Figura 2-22 - Simulação de circuito com diodo.

Para se alterar o valor dos componentes do circuito, das condições de simulação e dos parâmetros do diodo, basta carregar no circuito com o botão esquerdo do rato para que surjam as janelas relativas a estas configurações - Figura 2-23.

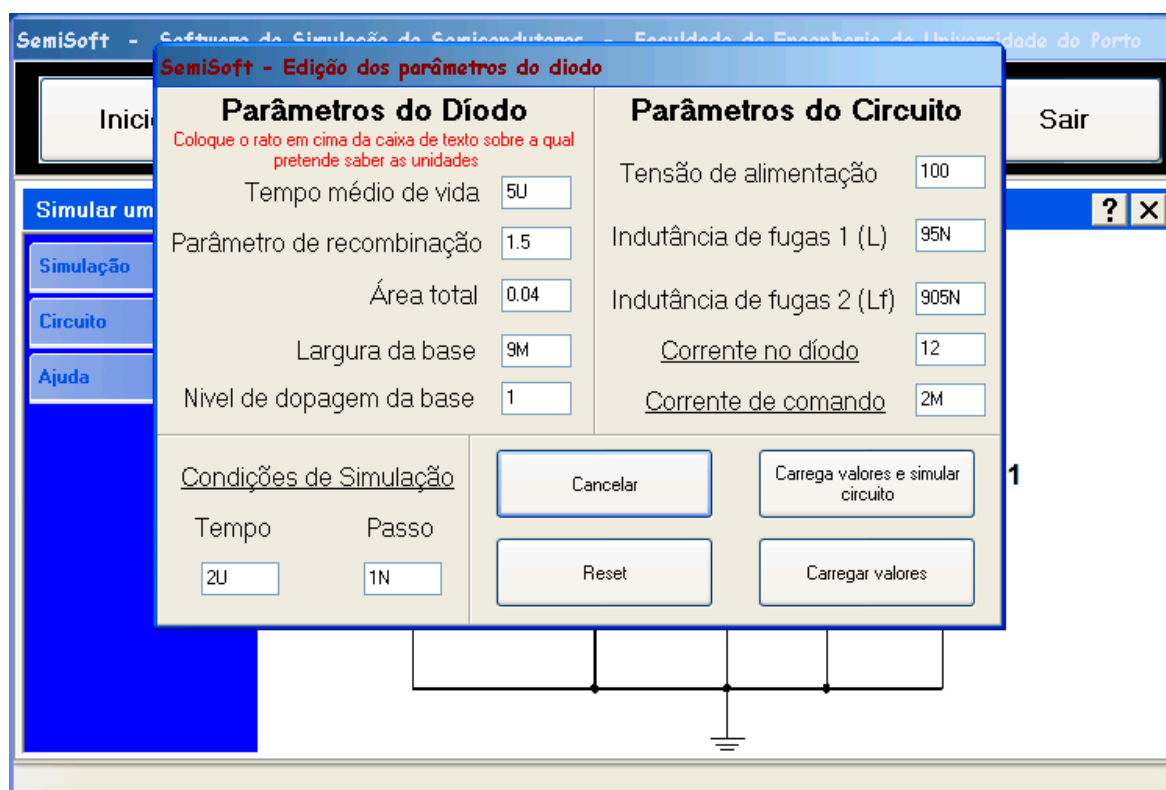


Figura 2-23 - Janela de parâmetros do diodo, circuito e simulação.

No que diz respeito aos gráficos, esta ferramenta permite desenhar gráficos a duas e três dimensões. Os gráficos 2D apresentam as formas de onda da tensão e da corrente em função do tempo. De notar que é possível visualizar o resultado de várias simulações, num máximo de sete, no mesmo gráfico. A Figura 2-24 permite visualizar esta mesma característica. Nas Figura 2-24 e Figura 2-25 é possível verificar que o programa não se ajusta à resolução de ecrã do computador do utilizador, dando a sensação que a imagem foi indevidamente cortada, embora não tenha sido esse o caso.

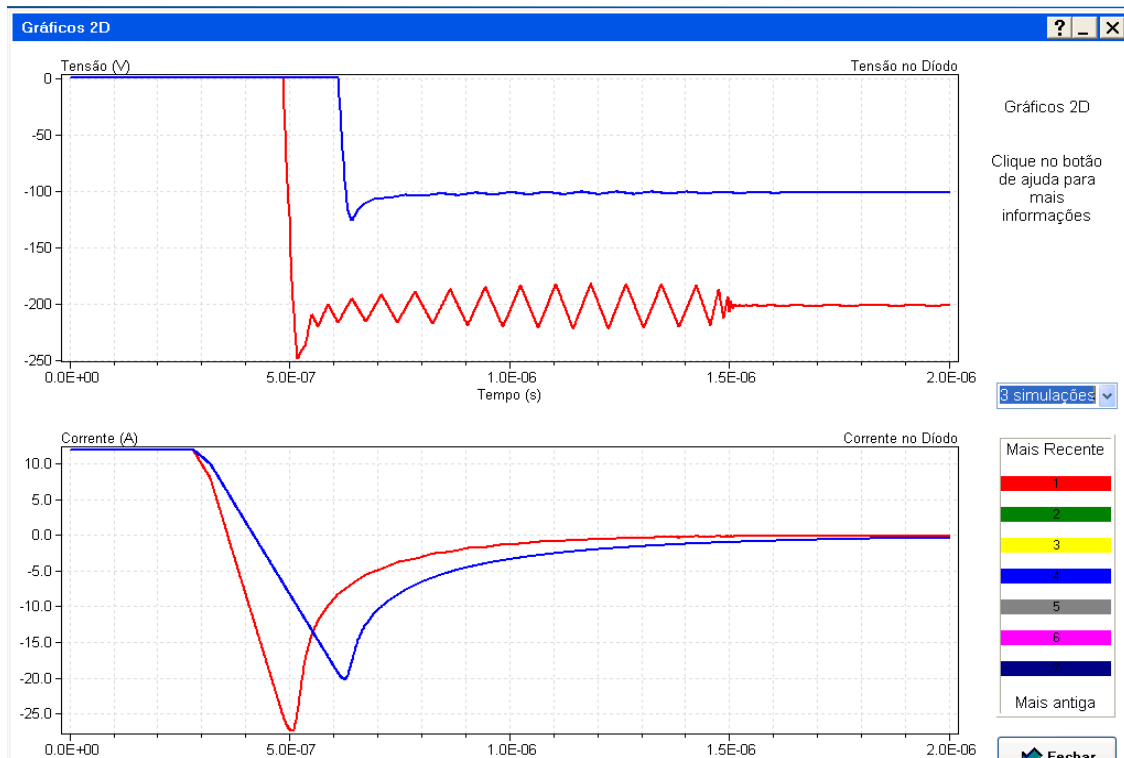


Figura 2-24 - Gráficos a duas dimensões da tensão e corrente no diodo.

Em relação aos gráficos a três dimensões, é apresentada uma superfície que relaciona os valores da concentração de electrões/lacunas com a largura da zona de armazenamento de carga (n^-) em função do tempo. No entanto, esta representação apresenta limitações importantes. Uma primeira, em termos visuais, na medida em que o eixo que representa a concentração (ZZ) não é perpendicular ao plano XY, podendo causar dificuldades na interpretação do gráfico. A segunda limitação relaciona-se com o facto de não ser possível a caracterização numérica dos eixos e a respectiva manipulação das escalas.

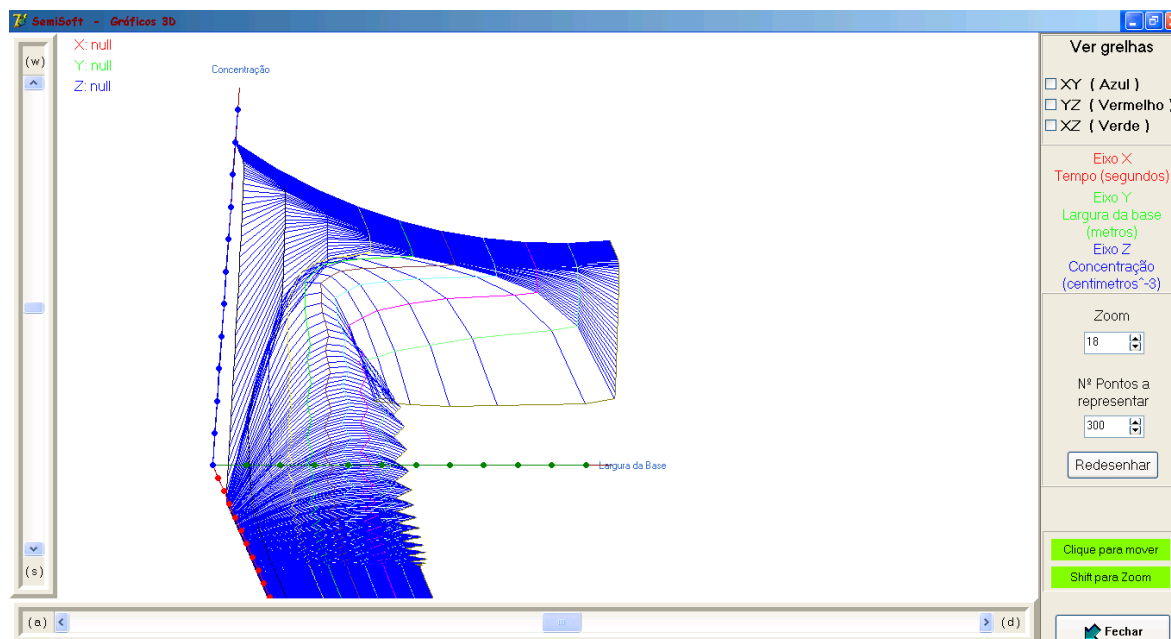


Figura 2-25 - Representação a três dimensões -SEMISOFT.

Outra potencialidade deste programa é o facto de permitir que o utilizador modifique os caminhos da pasta de histórico das simulações, do ficheiro de execução da aplicação e do simulador utilizado (Figura 2-26).



Figura 2-26 - Menu de configuração dos caminhos do sistema -SEMISOFT.

2.4 - O simulador PSIM®

Nesta secção apresenta-se um programa de simulação de circuitos de potência utilizado em larga escala nos meios académicos - o PSIM®. De referir que apesar de existirem outros programas de simulação, tais como NI Multisim®, Saber® e Simulink/Matlab®, optou-se por apresentar o PSIM®, uma vez que é um programa com o qual o autor da presente dissertação se familiarizou ao longo do seu percurso académico, para além de estar bastante orientado para aplicações na área da electrónica de potência [12].

O PSIM® é um programa de simulação especialmente projectado para electrónica de potência e controlo de motores. Este programa é composto por três componentes: desenho, simulação e visualização dos resultados em forma de gráficos bidimensionais. A interface gráfica de edição de circuitos é bastante intuitiva e a disponibilidade dos elementos mais utilizados em circuitos de potência e de controlo de motores permite que rapidamente se desenhe qualquer tipo de circuito. A Figura 2-27 apresenta o ambiente de desenho de circuitos do PSIM®.

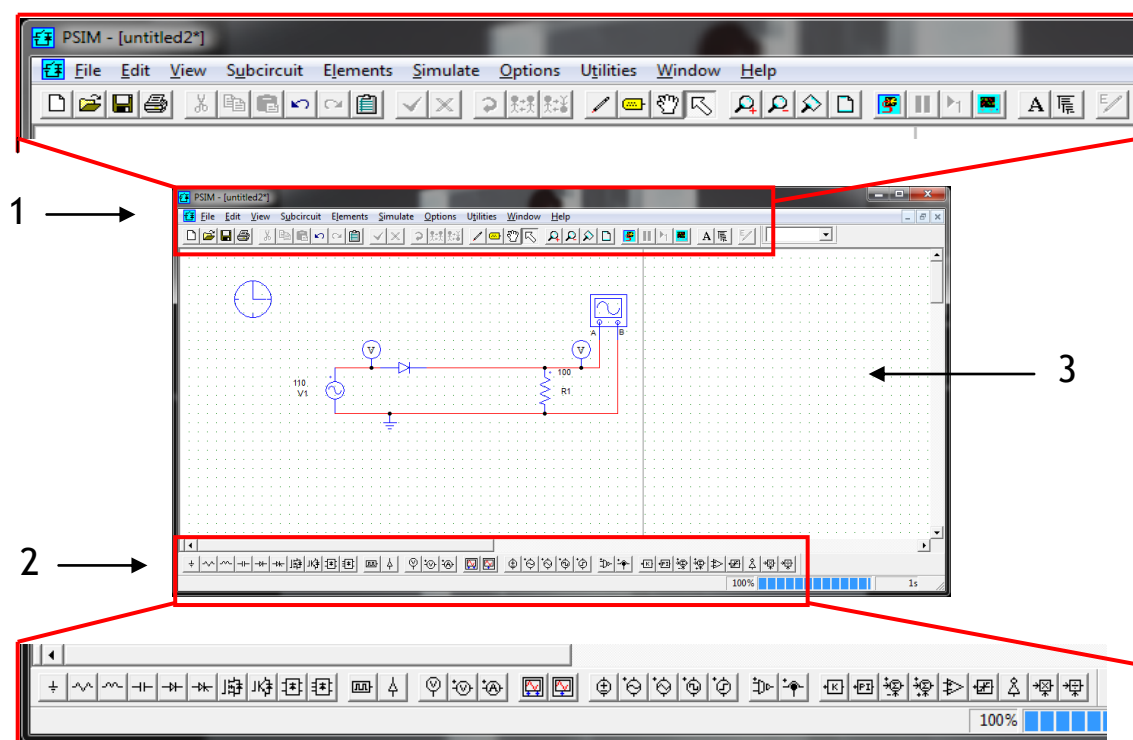


Figura 2-27 - Ambiente de desenho do PSIM®. 1- Barra de atalhos; 2-Barra de atalho dos componentes mais utilizados; 3- Desenho do circuito.

As barras de atalhos, assinaladas com os números 1 e 2 na Figura 2-27, são extremamente úteis e permitem melhorar a rapidez e ergonomia de utilização do programa. A barra dos

componentes permite que rapidamente se construa um circuito sem necessidade de procurar os componentes numa lista completa dos mesmos. Relativamente à área de desenho, o PSIM® tem uma função que permite representar partes de circuitos em blocos, ou seja, em sub-circuitos, aumentando assim a capacidade de desenhar circuitos complexos e de grande tamanho. Ao carregar num determinado componente, surge uma janela onde é possível alterar o valor, a cor, legenda, entre outras propriedades - Figura 2-28. Todavia, é de notar que neste programa não é possível modificar características físicas dos semicondutores. Desta forma, não é possível alterar parâmetros característicos de um diodo, tais como, o tempo média de vida, largura da e nível de dopagem da base, parâmetro de recombinação e área total.

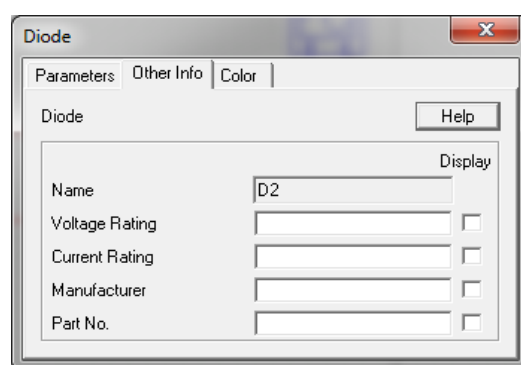


Figura 2-28 - Janela para modificar características do componente (diodo).

Relativamente à componente de simulação do PSIM®, pode-se dizer que este programa utiliza algoritmos eficientes para superar os problemas de convergência e de tempos de simulação elevados. O utilizador pode ver em que ponto vai a simulação através da informação gráfica e textual que aparece no canto inferior direito da barra de atalhes dos componentes (2) apresentada na Figura 2-27.

Os resultados da simulação são apresentados na janela de visualização dos resultados (Simview)- Figura 2-29. Ainda nesta figura, a barra legendada com o número 1 é uma barra de atalhos e permite aceder a várias funcionalidades de uma forma simples e intuitiva. Assim, é possível fazer zoom, mudar as escalas dos eixos, adicionar gráficos à janela de visualização, inserir legendas, etc. A barra marcada com o número 2 é uma barra extremamente útil porque permite encontrar os valores máximos/mínimos absolutos e relativos de forma simples e célere. Ao nível da área disponível para a representação de gráficos, é possível ao utilizador adicionar mais do que uma forma de onda no mesmo gráfico, ou então em gráficos separados.

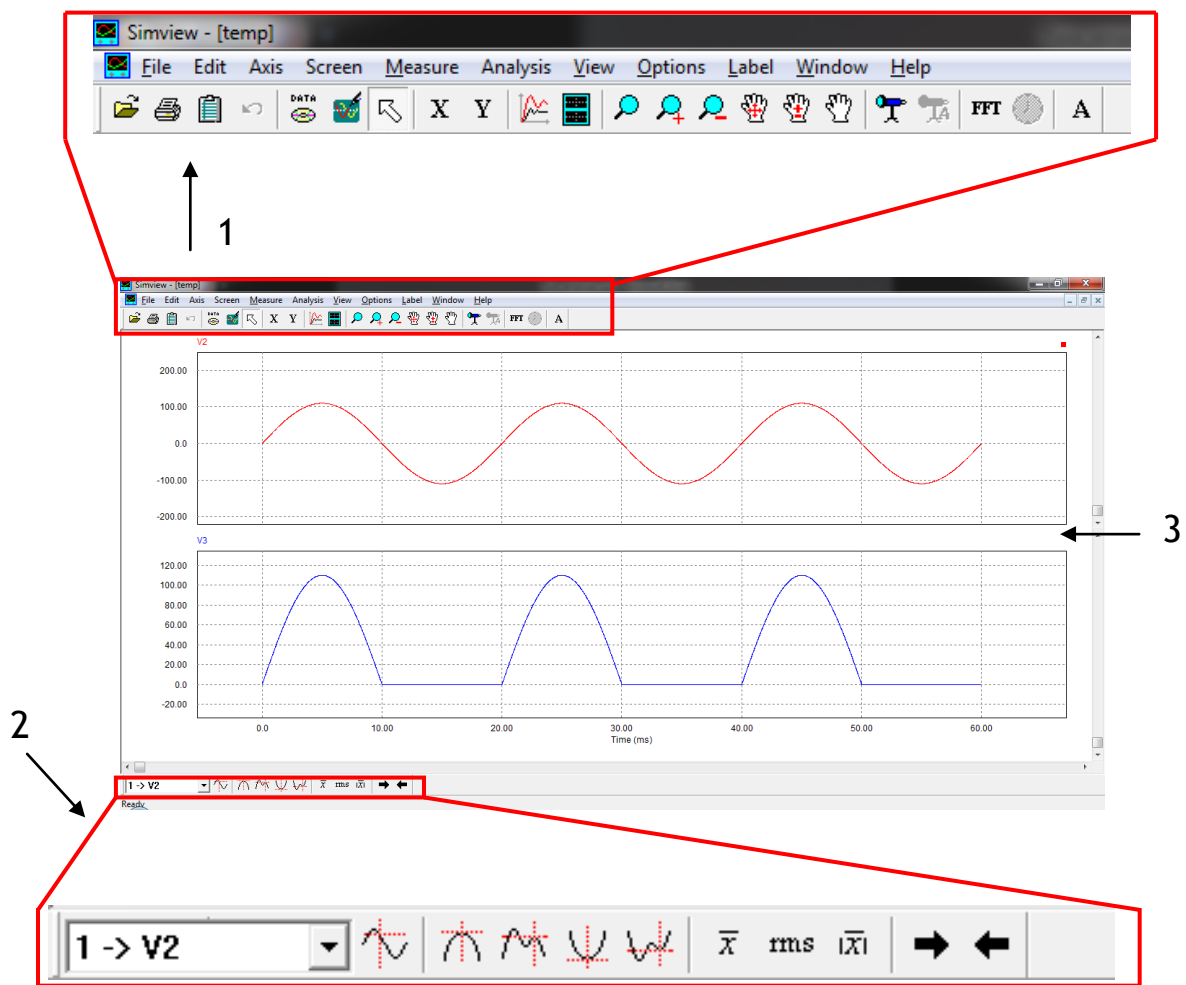


Figura 2-29 - Janela de visualização dos dados. 1- Barra de atalhos, 2- Barra de máximos e mínimos locais e globais, 3- Visualização dos gráficos.

Em suma, pode-se destacar o facto desta ferramenta de simulação de circuitos permitir desenhar qualquer tipo de circuito de potência, ou de comando de motores, com uma rapidez de simulação assinalável e elevada qualidade de apresentação de resultados. Deve-se também realçar que o PSIM® não permite alterar as propriedades físicas dos semicondutores e visualizar os efeitos destas alterações no comportamento dinâmico do circuito. Outra nota de relevo deixa-se a nível dos exemplos disponibilizados pela ferramenta, uma vez que não estão acompanhados de explicações sobre os resultados previstos e sobre os fenómenos que ocorrem.

2.5 - Conclusões

A história dos dispositivos electrónicos baseados em semicondutores é bastante interessante e longa, embora incida num curto espaço temporal. A área e o mercado da electrónica participam indirectamente ou directamente em todas as actividades humanas, sendo que, caso não se desse o constante avanço da electrónica, não seria possível avançar

nas diversas áreas de conhecimento humano. Desde a sua vida pessoal até à actividade profissional, o Homem está em contacto directo com diversos bens essenciais ao seu quotidiano e bem-estar, bens esses que, sem o avanço da electrónica, não existiriam. Como exemplo elucidativo da afirmação anterior é o simples despertador que faz com que a maioria das pessoas acorde à hora estipulada. A descoberta do transistor é o marco mais importante da história dos dispositivos electrónicos que utilizam materiais semicondutores, não só pelo seu contexto histórico (Fim da SGM) mas também pela sua importância no mundo da electrónica.

No que respeita aos programas educacionais para o ensino de semicondutores, o TESE e o SEMISOFT disponibilizam funcionalidades que viabilizam a sua utilização nas salas de aulas, permitindo aos estudantes alterar parâmetros físicos dos dispositivos electrónicos e visualizar a sua influência no comportamento dinâmico do circuito. Uma funcionalidade a destacar é a de desenho de gráficos 3D, no SEMISOFT, que permite visualizar a concentração de electrões em função do tempo e da largura da base do dispositivo. Contudo, estes programas ainda carecem de exemplos pré-definidos com explicações teóricas sobre o comportamento típico dos dispositivos electrónicos em função da alteração dos seus parâmetros físicos.

Relativamente ao programa de simulação de circuitos apresentado, o PSIM, é de realçar a sua velocidade de simulação, a facilidade de alterar valores dos componentes do circuito e a componente gráfica de apresentação dos resultados. Deve-se salientar ainda que este programa comercial também não tem presente nos seus exemplos pré-definidos explicações sobre os fenómenos físicos e eléctricos mais comuns.

Capítulo 3

Modelos dos semicondutores em estudo

3.1 - Introdução

Neste capítulo são abordados os modelos dos semicondutores (Díodo e IGBT) utilizados na simulação dos circuitos disponíveis na aplicação desenvolvida.

Na primeira secção deste capítulo faz-se uma apresentação sucinta dos referidos modelos e mostra-se a diferença entre estes e os modelos tipicamente disponíveis em ferramentas comerciais de simulação de circuitos.

De seguida são apresentados os circuitos de teste utilizados na aplicação desenvolvida, a explicação do seu funcionamento e os parâmetros acessíveis a modificações.

3.2 - Síntese dos modelos dos semicondutores utilizados

Os modelos dos dispositivos semicondutores utilizados na interface para simulação de circuitos baseiam-se nas propriedades físicas destes, sendo que o modelo do díodo foi desenvolvido por A. Araújo [6] e o modelo do IGBT por R. Chibante [41] .

Os díodos são elementos semicondutores construídos pela junção de dois materiais semicondutores, um do tipo n e outro do tipo p, formando assim uma estrutura PN que, dentro dos seus limites de tensão e de corrente, permitem a passagem da corrente apenas num sentido. A Figura 3-1 apresenta a estrutura de um díodo semiconductor.

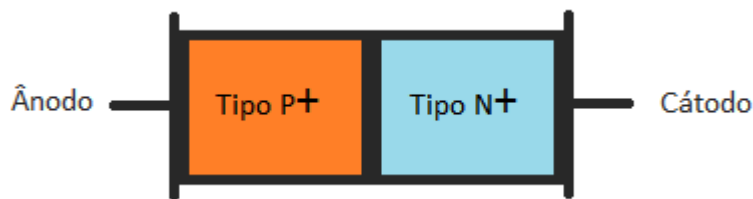


Figura 3-1 - Esquema de um diodo convencional.

O modelo desenvolvido para o diodo refere-se a um diodo de potência e desta forma a sua estrutura é ligeiramente diferente da de um diodo convencional (Figura 3-1) de forma a poder suportar tensões mais elevadas - Figura 3-2.

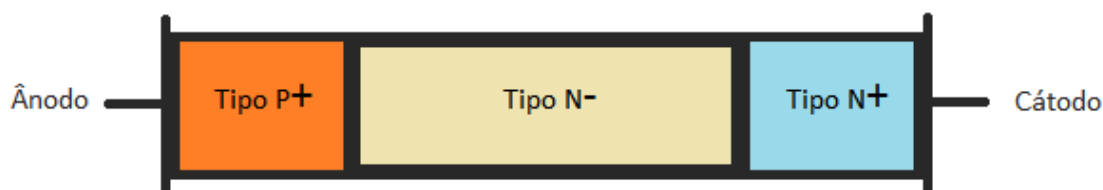


Figura 3-2 - Estrutura típica de um diodo de potência.

A introdução da região com baixa dopagem, tipo N^- provoca a diminuição do campo eléctrico na transição, dando uma forte característica resistiva quando em condução. Esta característica torna-se mais significativa quando a tensão se aproxima do valor máximo suportado pelo dispositivo. Da análise da Figura 3-2 pode-se identificar três zonas distintas: Tipo P^+ , Tipo N^- , Tipo N^+ .

No que diz respeito ao IGBT modelado, a sua estrutura é apresentada na Figura 3, sendo possível identificar 4 zonas.

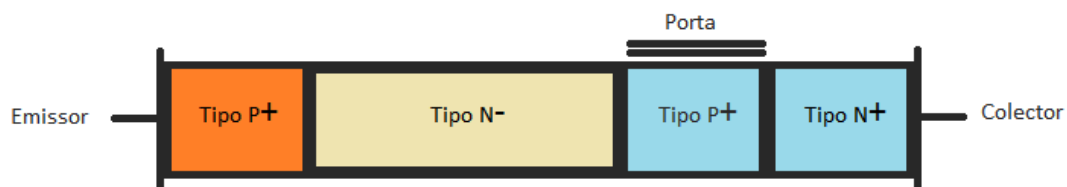


Figura 3-3 - Estrutura típica de um IGBT.

As diferentes zonas que constituem os semicondutores de junção bipolar são modeladas por processos diferentes. Assim, os modelos dos semicondutores utilizados nesta dissertação caracterizam-se por apresentarem uma abordagem híbrida, ou seja, a zona N^- é modelada tendo em conta as suas características físicas enquanto as restantes zonas são modeladas por processos empíricos.

A base dos modelos desenvolvidos reside na resolução da equação de difusão ambipolar (EDA) que permite conhecer a distribuição dos electrões/lacunas em função do tempo e da largura da zona N⁻. A abordagem de resolução parte de uma formulação variacional, seguida por uma aproximação por elementos finitos [6]. Esta abordagem permite que sejam implementados modelos baseados nas características físicas dos semicondutores em simuladores do tipo SPICE, através de uma analogia entre o sistema de equações diferenciais ordinárias, associadas à resolução da EDA, e o sistema de equações associado a um circuito composto por resistências e condensadores (Figura 3-4)

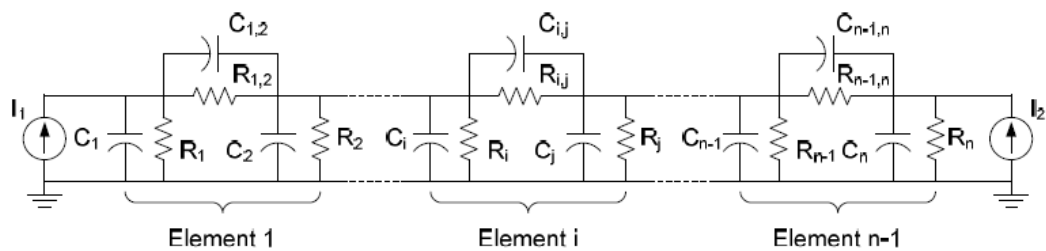


Figura 3-4 - RC para a solução da EDA para elementos lineares [6].

Com esta abordagem híbrida é possível descrever o comportamento dinâmico e estático dos dispositivos semicondutores em causa com boa precisão, na medida em que tal descrição é baseada no conhecimento da distribuição temporal e espacial dos portadores de carga, mantendo tempos de simulação da ordem de poucos segundos.

3.3 - Circuitos de Teste

Nesta secção são apresentados os circuitos de teste para o díodo e para o IGBT disponíveis na aplicação desenvolvida. Estes circuitos serão usados para demonstrar as funcionalidades e potencialidades pedagógicas da aplicação.

3.3.1 - Díodo

O circuito de teste do díodo é apresentado na Figura 3-5 e o seu funcionamento é explicado de seguida:

- Com a entrada em condução do transistor, T1, estabelece-se um valor bem definido de corrente em L1 e LF.
- A abertura do transistor, T1, impõe a condução do díodo de teste (DUT) em conjunto com L1 e L.
- Após o estabelecimento da corrente, um novo fecho do transistor (T1) impõe uma descida da corrente no díodo (DUT) com um gradiente de $(-V1)/(L + LF)$. Deve-se

salientar que o valor da bobine L1 deve ser escolhido para que, durante o tempo de comutação, se comporte como uma fonte de corrente constante.

- A partir do momento em que, durante a comutação, o gradiente da corrente no díodo, DUT, se inverte, o díodo D1 entra em condução.
- Assim, durante o tempo de recuperação inversa, o gradiente da corrente é controlado pela bobine LF.
- A recuperação após o pico de corrente inversa e o restabelecimento do poder de bloqueio do díodo é feito sobre a bobine L.

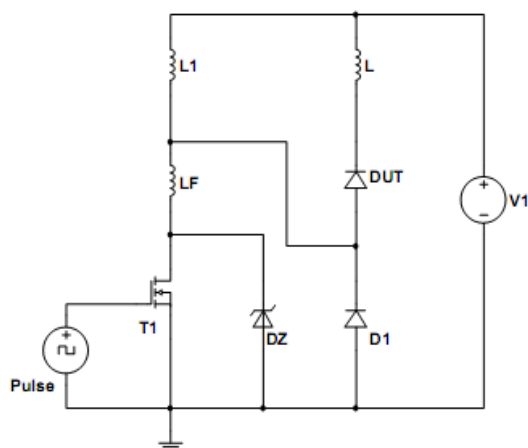


Figura 3-5 - Circuito de teste do díodo.

Como já foi referido, o modelo do díodo permite alterar os parâmetros físicos deste. A Tabela 3-1, apresenta os parâmetros envolvidos, a sua descrição e a sua grandeza.

Tabela 3-1 - Parâmetros do modelo do díodo.

Símbolo	Grandeza	Descrição
A	cm ²	Área do semicondutor
Wb	cm	Comprimento da Base
NB	cm ⁻³	Concentração por volume da base
Hp	cm ⁴ /s ⁻¹	Parâmetro de recombinação
T	s	Tempo médio de vida

Além dos parâmetros físicos do semicondutor, é possível alterar parâmetros do circuito para se testarem diferentes configurações. De seguida faz-se uma breve apresentação dos restantes componentes:

- V1 - Fonte de alimentação do circuito;
- L - Indutância de fugas 2;
- LF - Indutância de fugas 1;
- L1 - Indutância principal (comportamento como fonte de corrente constante);
- Pulse - Tensão de controlo;

3.3.2 - IGBT

O circuito de teste utilizado para o IGBT -Figura 3-6, é composto por uma carga resistiva (R_L) com indutância de fugas (L_c) e por um circuito de comando puramente resistivo e o seu funcionamento é explicado de seguida:

- A velocidade de comutação do IGBT é controlada pelo valor da resistência no circuito de comando. Assim, quanto menor a resistência maior será a velocidade de entrada/saída de condução do IGBT.
- Esta configuração é particularmente sensível à indutância de fugas do circuito, especialmente na saída de condução do IGBT. Como a queda de tensão na indutância é proporcional à taxa de variação da corrente, e esta é bastante elevada durante a comutação, a tensão colector-emissor pode apresentar sobre-tensões bastante significativas relativamente à tensão de bloqueio.

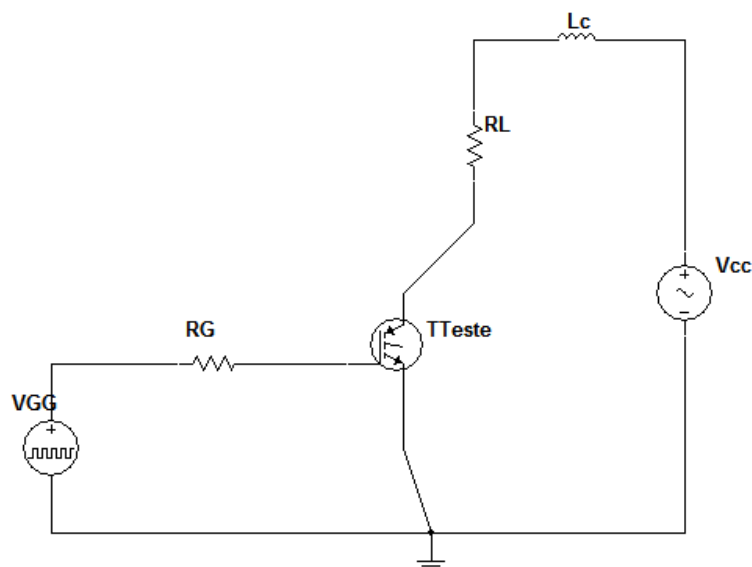


Figura 3-6 - Circuito de teste do IGBT (Tteste).

A Tabela 3-2 apresenta os parâmetros do modelo do IGBT, a sua grandeza e descrição. Todos estes parâmetros podem ser alterados na aplicação desenvolvida, sendo assim possível analisar a sua influência no comportamento estático e dinâmico do circuito.

Tabela 3-2- Parâmetros do modelo IGBT.

Símbolo	Grandeza	Descrição
A	cm^2	Área total
Agd	cm^2	Área gate-dreno
Cgs	F	Capacidade gate-fonte
Coxd	nF	Capacidade de depleção óxido gate-dreno
Hp	$10^{-14} \text{ cm}^4/\text{s}$	Parâmetro de recombinação do emissor
Kf	-	Factor de transcondutância da região de tródo MOS
Kp	A/V^2	Transcondutância da região de saturação MOS
NB	10^{14} cm^{-3}	Concentração na base (nível de dopagem)
Vth	V	Tensão limiar de condução
WB	cm	Largura metalúrgica da base
T	s	Tempo médio de vida

Além dos parâmetros físicos do semiconductor, é possível alterar parâmetros do circuito para se testarem diferentes configurações.

De seguida faz-se uma breve descrição dos restantes dispositivos:

- VCC - Fonte de Alimentação do circuito;
- VGG - Tensão de Controlo;
- LC - Indutância de Fugas;
- RC - Resistência de Carga
- RG - Resistência da Gate;

3.4 - Conclusões

Neste capítulo apresentaram-se sumariamente os modelos dos díodos e IGBTs utilizados na simulação de circuitos. A sua principal característica é terem uma base física importante, possibilitando estudar a influência de características físicas e tecnológicas de um semicondutor, bem como a influência dos componentes que constituem um dado circuito.

Capítulo 4

Ferramentas de programação utilizadas

4.1 - Introdução

Este capítulo apresenta as ferramentas de programação utilizadas no desenvolvimento da interface, o ICAPS® e o DELPHI BORLAND 7®. [42]

O programa ICAPS® é responsável pela simulação dos circuitos utilizados na interface. Este programa é composto por quatro módulos: esquemático, editor de texto, simulador e visualizador dos resultados de simulação. O editor de texto e o simulador são, entre os módulos apresentados, aqueles que vão ser utilizados pela interface desenvolvida para a simulação dos circuitos.

Esta ferramenta disponibiliza toda a informação (disposição dos componentes, resultados da simulação, erros de simulação, etc.) sobre o circuito simulado em ficheiros de texto, facilitando, por exemplo, a pesquisa e recuperação dos resultados por outra aplicação distinta.

Para desenhar e desenvolver a interface foi necessário escolher uma ferramenta de desenvolvimento de aplicações, facto pelo qual se realizou um levantamento de algumas alternativas possíveis, sendo que a decisão final recaiu sobre o DELPHI BORLAND 7®. Assim, na secção 4.3 apresenta-se esta ferramenta, bem como a linguagem de programação que utiliza e um pequeno exemplo sobre o seu funcionamento.

4.2 - A ferramenta de simulação ICAPS®

4.2.1 - Esquemático

O módulo esquemático é responsável pela criação e edição gráfica de circuitos através do “SpiceNet”. Este contém as ferramentas essenciais para o projecto de circuitos e possui algumas particularidades que facilitam o seu desenho e edição, tais como a edição dos modelos dos componentes utilizados, criação de novos modelos para os componentes, a implementação de pequenos circuitos com apenas alguns terminais acessíveis (sub-circuitos), entre outras. Para exemplificar as potencialidades do ICAPS®, pode-se utilizar um modelo de um dispositivo semiconductor (diodo) e modificar o valor das suas características físicas, (tempo média de vida, largura e nível de dopagem da base, parâmetro de recombinação e área total), analisando-se, posteriormente, o comportamento do circuito e do dispositivo.

O exemplo apresentado na Figura 4-1 ilustra o ambiente de edição e desenho de circuitos do ICAPS® - o SpiceNet.

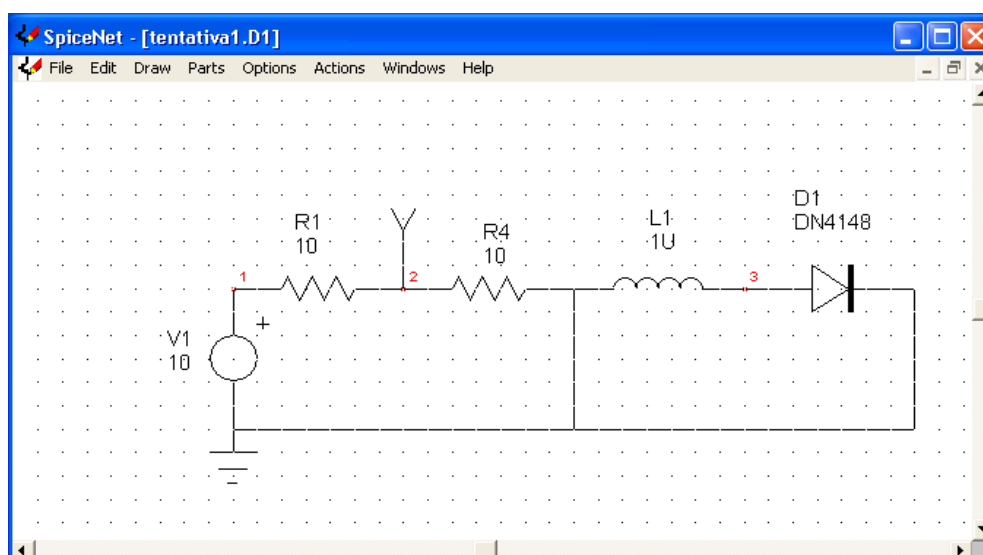


Figura 4-1- Exemplo do desenho de um circuito no ICAPS.

O ambiente de desenvolvimento do ICAPS® é pouco intuitivo e ergonómico no que toca à sua utilização. A ausência das barras de atalhos com os componentes característicos de um circuito electrónico implica que seja necessário procurar o componente desejado numa lista, gastando-se tempo que poderia ser direccionado para o desenho do circuito em si.

Para concluir a apresentação deste módulo, de referir que o ICAPS® guarda a informação do circuito desenhado num ficheiro com a extensão “.cir”. Na secção 5.9 do capítulo 5 é feita uma explicação das extensões dos ficheiros e da sua utilidade no desenvolvimento da interface.

4.2.2 - Editor de texto

A ferramenta de simulação, ICAPS®, tal como a maioria dos simuladores de circuitos electrónicos, guarda a informação do circuito implementado através de uma estrutura de nós que se interligam entre si através dos componentes. Essa informação é guardada como ficheiro de texto, havendo a possibilidade de o editar, caso o utilizador pretenda. Desta forma, a alteração de parâmetros do circuito pode ser realizada sem recorrer ao modo gráfico. Este módulo é denominado por “IsEd4” e é apresentado na Figura 4-2. Esta figura mostra o ficheiro de texto com a informação relativa ao circuito da Figura 4-1.

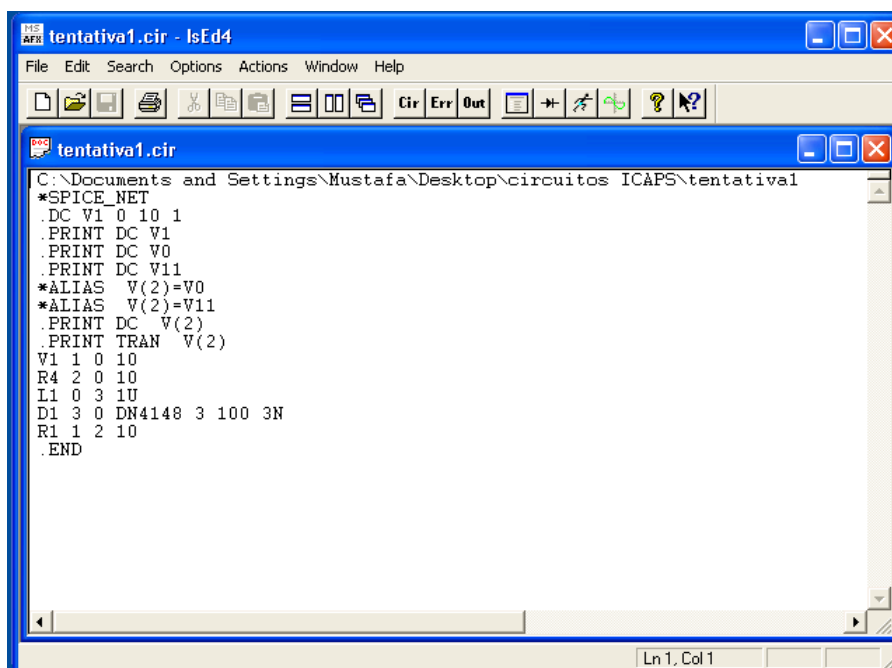


Figura 4-2- Exemplo da edição de um circuito, utilizando o módulo de editor de texto, IsEd4.

Da análise do ficheiro de texto apresentado na figura anterior, depreende-se que este é constituído por várias linhas, tendo cada uma a sua função. Por exemplo, considere-se a linha com o código “V1 1 0 10”. Este código significa que a fonte de tensão com o nome “V1” está ligada entre os nós “1” e “0” e fornece o valor de “10” Volt.

Encontram-se ainda, noutras linhas do mesmo ficheiro, as seguintes expressões: “*PRINT DC*”, “*PRINT TRAN*” e “*ALIAS*”. As duas primeiras são usadas para guardar os dados resultantes da simulação em análise contínua (“*PRINT DC*”) ou em análise transitória (“*PRINT TRAN*”). A terceira expressão é usada para definir nomes alternativos (“V11”) para uma dada variável (“V(2)”).

De referir ainda que a complexidade deste ficheiro de texto depende do número de componentes do circuito, do número de variáveis que se pretende guardar e da natureza dos componentes utilizados (lineares ou não lineares).

4.2.3 - Simulador

O módulo de simulação no ICAPS® denomina-se IsSpice e é composto por quatro janelas, como é visível na Figura 4-3. A janela de “*Simulation Status*”, assinalada na figura com o número 1, permite visualizar os resultados à medida que a simulação vai decorrendo. A janela marcada com o número 2 é a janela de “*Errors and Status*” e exibe mensagens com os possíveis erros que podem ocorrer durante a simulação. Existe ainda uma outra janela, “*Output*”, assinalada com o número 3, que exibe scripts², no caso de estes terem sido adicionados ao circuito aquando da sua edição. Por fim, a quarta janela, que compõe o módulo IsSpice, marcada com o número 4, “*Simulation Control*”, surge apenas quando a simulação termina, permitindo alterar o tipo de análise seleccionada, implementar scripts, adicionar expressões, ect. Quando esta janela aparece, os dados resultantes da simulação são guardados num ficheiro de texto, com a extensão “.out”.

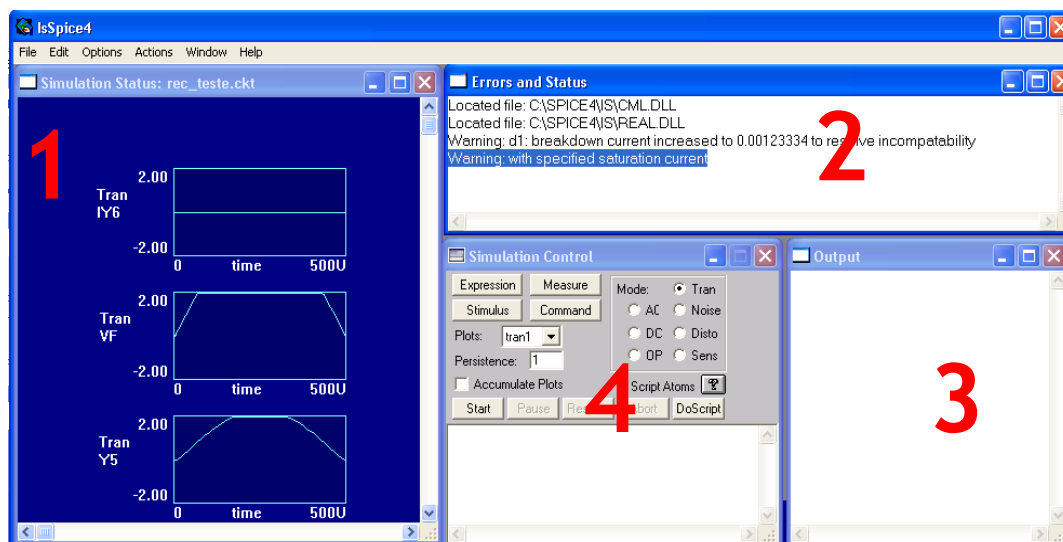


Figura 4-3 - Janela de Simulação “IsSpice4”. 1 - “Simulation Status” ; 2 - “Errors and Status”, 3 - “Output”, 4 - “Simulation Control”.

Para efectuar uma simulação é necessário fornecer ao simulador um ficheiro com a extensão “.ckt”. Este ficheiro é gerado pelo editor de texto ou pelo editor gráfico e difere muito pouco do ficheiro com a extensão “.cir”, apresentado anteriormente. No entanto, apenas o ficheiro “.ckt” possui todos os circuitos e subcircuitos, com as respectivas bibliotecas na forma standard de simulação.

O módulo de simulação, IsSpice4 tem a propriedade de poder ser executado independentemente do ICAPS®. Para tal basta usar a seguinte linha de comandos:

² Um script é um programa ou uma sequência de instruções que são interpretadas ou executadas por outro programa e não pelo processador.

[caminho da pasta do executável “SPICE4.exe”] + [espaço] + [caminho do ficheiro “.ckt”]

De salientar que esta propriedade vai ser usada no desenvolvimento da interface educativa para o ensino de semicondutores, permitindo que as simulações sejam efectuadas sem que seja necessário recorrer ao ICAPS®.

4.2.4 - Visualizador de Sinais

Este módulo, designado por “IntuScope”, representa graficamente a informação gerada na simulação. Porém, só permite visualizar gráficos bidimensionais. O *layout* é composto por cinco janelas, podendo-se tornar algo confuso para um utilizador inexperiente, como é perceptível na Figura 4-4.

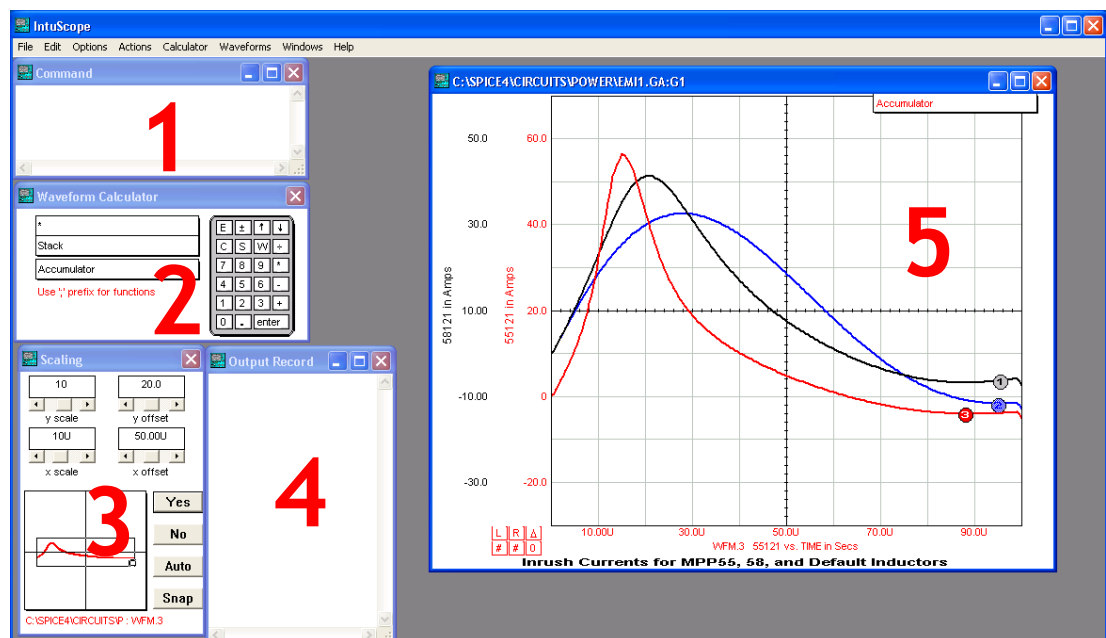


Figura 4-4 - Módulo de visualização dos sinais - IntuScope. 1 - “Comand”, 2 - “Waveform Calculator”, 3 - “Scaling”, 4 - “Output Record”, 5 - “Graph”.

As janelas que compõem o módulo de visualização dos resultados estão numeradas e são explicadas de seguida:

1. “Comand” - permite introduzir uma série de acções de controlo que são executadas como scripts;
2. “Waveform Calculator” - possibilita utilizar dados provenientes dos gráficos para realizar cálculos com números reais e complexos;
3. “Scaling” - facilita a visualização dos dados através da alteração das escalas dos eixos;
4. “Output Record” - permite visualizar os scripts adicionados;
5. “Graph” - visualização dos gráficos e das formas de onda;

Por último, referir ainda que este módulo não permite correlacionar a evolução de várias variáveis a três dimensões.

Como conclusão, pode-se dizer que o ICAPS®:

- Apresenta uma interface pouco intuitiva e complicada de usar;
- Possibilita a execução independente de cada um dos seus módulos;
- Permite a criação de novos modelos de dispositivos electrónicos e a edição dos já existentes;
- Possibilita a edição de circuitos através da manipulação de um ficheiro de texto.

4.3 - Ferramenta de Programação (Delphi 7)

4.3.1 - Tomada de decisão

A discussão em torno da ferramenta que melhor se adaptava aos requisitos e objectivos propostos no Capítulo 1 para desenvolvimento do programa para o ensino de semicondutores, motivou que se procedesse a um levantamento de possíveis soluções existentes no mercado para o desenvolver. Das soluções encontradas destacaram-se as seguintes: Borland Delphi 7®, Lazarus Free Pascal®, Matlab®, .Net® e Visual Basic®.

Com a finalidade de facilitar a escolha da ferramenta de desenvolvimento, definiu-se uma tabela, Tabela 4-1, com os seguintes critérios: experiência do autor com a ferramenta em causa, domínio da linguagem de programação associada, ajuda online, e geração de um ficheiro executável. Utilizando uma escala de 0 a 5 valores, pontuaram-se estes critérios, tendo a escolha recaído sobre o Borland Delphi 7®.

Tabela 4-1 - Critérios de selecção da ferramenta gráfica a utilizar.

Ferramenta Gráfica	Experiência do autor	Gera Ficheiro Executável	Linguagem Programação Conhecida	Ajuda Online	Escolha Final
Lazarus Free Pascal®	5	5	5	3	2
.Net®	0	5	0	5	4
Matlab®	3	0	3	5	3
Borland Delphi 7®	5	5	5	5	1
Visual Basic®	0	5	0	5	5

Embora se tenha optado por usar o programa Borland Delphi 7®, não se pode afirmar que este é melhor ou pior do que qualquer uma das outras opções, uma vez que todos possuem pontos negativos e positivos. *“No language is perfect nor perfectly suited for every programming purpose”*[43].

O Borland Delphi 7 é uma ferramenta que proporciona um ambiente de desenvolvimento gráfico com características próprias, das quais se podem distinguir as seguintes:

- Existência de um leque alargado de componentes, tais como botões, telas, painéis, editores de texto, gráficos, etc.;
- Existência de inúmeros aplicativos que permitem adicionar novos componentes, aumentando a quantidade e a qualidade dos já existentes de origem;
- Barras de atalhos que melhoram a navegação e escolha dos componentes;
- Utiliza como linguagem de programação o Object Pascal, que permite usar os conceitos de classe, polimorfismo, e herança. Estes conceitos são aprofundados na secção 5.3 do capítulo 5.
- Permite programação orientada a objectos e a eventos, ou seja, cada componente adicionado permite que lhe sejam adicionados eventos/acções.
- Geração de um ficheiro executável da aplicação desenvolvida que possibilita a sua utilização independentemente do programa de desenvolvimento.

4.3.2 - Linguagem de programação

A linguagem de programação que caracteriza o Delphi7® é a Object Pascal. Esta linguagem deriva da linguagem de programação Pascal, desenvolvida na década de 70 por Niklaus Wirth, introduzindo conceitos de programação orientada aos objectos [45][46]. O paradigma de programação orientada aos objectos oferece vantagens em relação às linguagens tradicionais, porque cada objecto possui dados e funções que podem ser reutilizadas. Desta forma, esta

linguagem é fortemente estruturada, permitindo que o programador divida o seu programa em pequenas partes (estruturas, classes), sendo que cada parte é responsável por uma determinada funcionalidade. No final, pretende-se que, com a interligação de todas as partes, a aplicação execute todas as funcionalidades pretendidas. Recorrendo a pequenas estruturas de dados é possível reutilizar partes de código implementado quando os objectos possuem as mesmas características.

A Object Pascal permite utilizar diversos tipos de dados (int, float, string, boolean, etc.) e diversas estruturas de controlo (for, if-then-else, while, case, etc), podendo-se encontrar com maior detalhe informação sobre a sintaxe de programação desta linguagem em [45].

4.3.3 - Exemplo de aplicação

No sentido de demonstrar as características apresentadas anteriormente, implementou-se um pequeno programa utilizando o Delphi7®. Este exemplo é apresentado na Figura 4-5, baseando-se no conhecido “Hello World”.

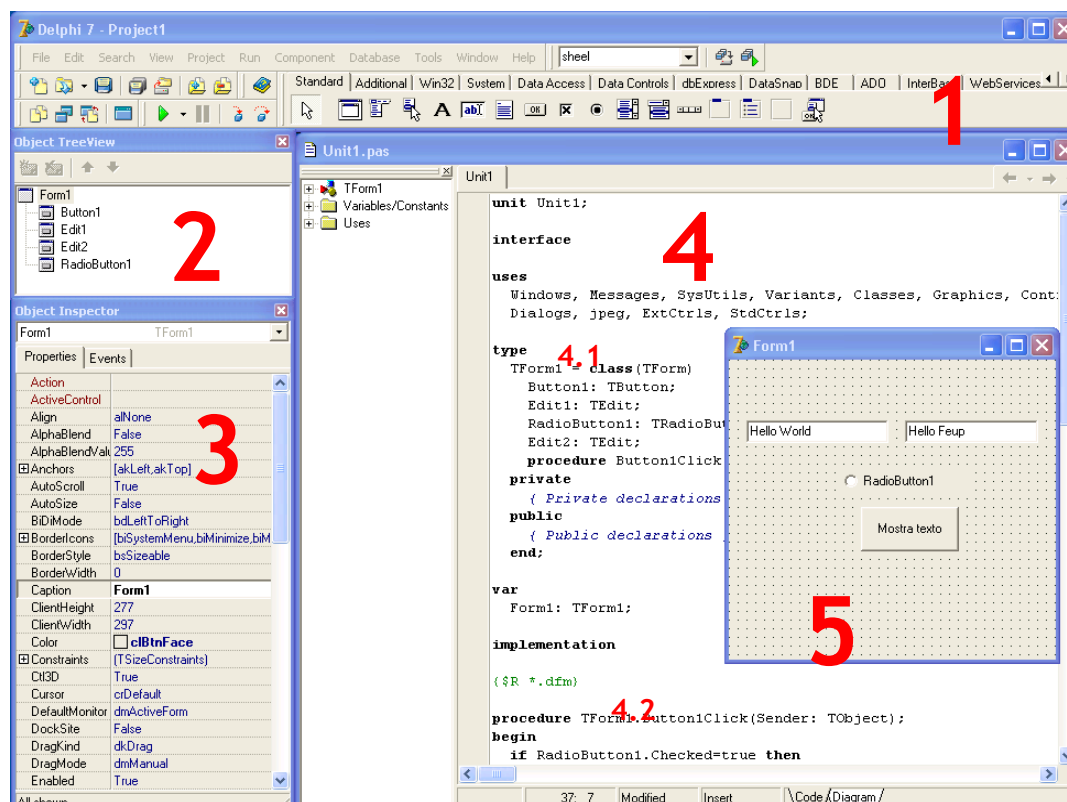


Figura 4-5 - Exemplo ilustrativo de um pequeno programa “Hello World”. 1 - Janela de atalhos. 2 - “Object TreeView”. 3 - “Object Inspector”. 4 - Janela de programação. 5 - “Form”.

As várias janelas que compõem o exemplo ilustrado anteriormente desenvolvido no ambiente gráfico Delphi 7®, estão numeradas e são apresentadas de seguida:

1. Barra de atalhos - Esta barra é composta pelos atalhos globais do sistema (abrir projecto, fechar e guardar projecto, iniciar, parar e cancelar simulação, etc) e por atalhos para inserção de componentes de desenho. Como se pode ver estes estão agrupados por abas, de acordo com o seu tipo.
2. “Object TreeView” - Esta janela permite listar todos os componentes utilizados no projecto e que são visíveis na “Form”.
3. “Object Inspector” - Esta janela possibilita que as características próprias dos componentes, tais como comprimento, largura, nome, legenda, possam ser editadas.
4. Janela de programação - Um programa pode ter várias janelas destas, denominadas “unit’s”, por defeito. A cada “unit” está, geralmente, associada a uma “Form” cujo todo o código que gera é disponibilizado na “unit”. Nesta janela é ainda possível visualizar, através do código, duas características referidas anteriormente: a utilização de classes, assinalada na Figura 4-5 com 4.1 e a programação orientada a eventos, 4.2.
5. “Form” - Aqui aparecem todos os componentes que forem utilizados na interface desenvolvida. No caso do exemplo apresentado, é utilizado um botão, duas caixas de texto e um “Radio button”.

Depois de compilar o programa desenvolvido e, caso não existam erros de sintaxe, de semântica ou de lógica, basta carregar na tecla F9 para executar a aplicação. Ao executar a aplicação surgirá uma janela, Figura 4-6, em tudo idêntica à janela “Form” apresentada anteriormente, só que em modo de execução.



Figura 4-6 - Janela do exemplo apresentado em execução.

O exemplo implementado é relativamente simples mas permite visualizar algumas propriedades do programa Delphi7®. Quando se carregar no botão “mostra texto”, o programa vai responder ao evento, com uma acção. Esta acção é definida pelo programador, sendo neste caso descrita da seguinte forma:

- caso o “Radio Button” não esteja seleccionado, o texto que deve aparecer é “Hello World”, ou seja, o da caixa de texto à esquerda .
- caso o “Radio Button” esteja seleccionado, o texto que deve aparecer é o da caixa de texto do lado direito, ou seja, “Hello Feup” ;

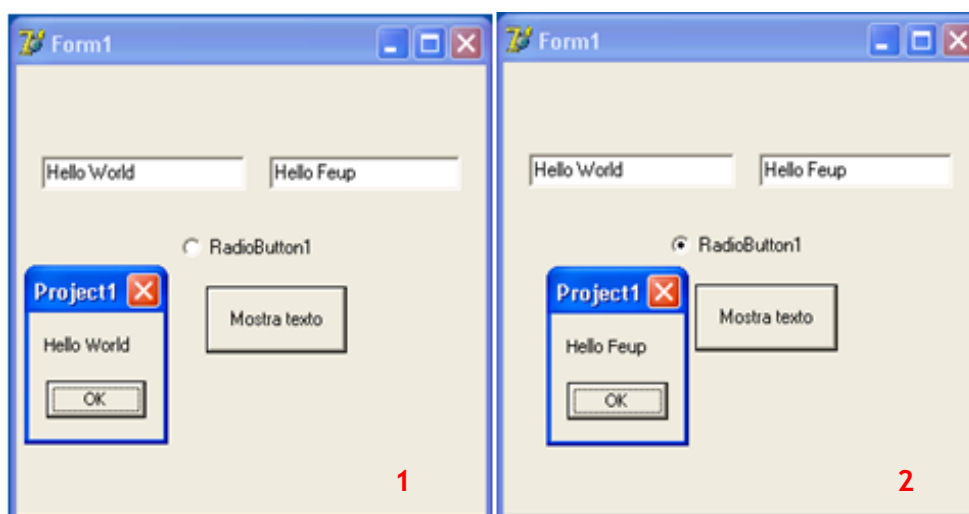


Figura 4-7 - 1- No caso de o “Radio Button” não estar seleccionado. 2- No caso de o “Radio Button” estar seleccionado.

4.4 - Conclusões

Neste capítulo começou-se por apresentar a ferramenta utilizada para a simulação de circuitos - o ICAPS®, tendo-se concluído que, embora seja uma ferramenta antiga e pouco intuitiva, tem a vantagem de permitir utilizar individualmente os módulos que o compõem.

Posteriormente, procedeu-se à escolha da ferramenta de desenho de interfaces e concluiu-se que o Borland Delphi 7® possibilita a concretização dos objectivos propostos.

A escolha desta ferramenta, em detrimento de outras, foi feita através da construção de uma tabela com alguns critérios definidos pelo programador da aplicação, tendo-se concluído que a ferramenta escolhida se adapta às necessidades.

Capítulo 5

Implementação da interface de simulação

5.1 - Introdução

A interface desenvolvida insere-se num grupo de aplicações denominadas por UI's (*User's Interfaces*) e que se caracterizam por estabelecerem uma relação entre estas e um ou vários utilizadores. Desta forma, a implementação de uma interface gráfica para o apoio do ensino da teoria associada aos dispositivos electrónicos semicondutores devem obedecer a várias regras de design e de concepção. Na secção 5.2 do presente capítulo apresentam-se os aspectos essenciais a considerar no desenvolvimento de uma interface com elevada versatilidade, ergonomia e usabilidade.

A qualidade desta interface não depende apenas do seu aspecto visual mas também da forma como o programador desenvolveu e estruturou o código da aplicação. Neste contexto, na secção 5.3, introduz-se a programação utilizando classes³, realçando-se as vantagens de as utilizar através de um pequeno exemplo ilustrativo.

As classes desenvolvidas e utilizadas na interface são descritas desde a secção 5.4. Optou-se por explicar individualmente e detalhadamente cada classe com o intuito de destacar, mais clarificadoramente, a importância de cada uma. Posteriormente, na secção 5.5, com recurso à linguagem de modelização UML (*Unified Modeling Language*) apresenta-se e explica-se a interacção entre as várias classes implementadas e o funcionamento global da interface.

Na secção 5.6, faz-se a exposição de algumas funções que se relevaram imprescindíveis para o cumprimento dos objectivos propostos.

Na secção 5.7 apresenta-se o componente extra utilizado no Delphi para as representações gráficas.

³ Uma classe é um grupo de objectos que possuem as mesmas características e os mesmos comportamentos.

Por fim, nas secções 5.8 e 5.9 apresenta-se o trabalho extra desenvolvido e que se afasta dos objectivos inicialmente propostos.

5.2 - Regras básicas para uma interface de sucesso

Um dos maiores desafios na implementação de uma interface gráfica passa por conseguir que esta seja simples, clara e consistente. Neste sentido, definiram-se, então, oito regras que, no caso de serem verificadas, melhoram a performance da aplicação desenvolvida. Estas oito regras [8] definidas são apresentadas de seguida:

- **Clareza** - Uma interface gráfica diz-se clara, ou que tem clareza, quando se consegue evitar ambiguidades, ou seja, a informação transmitida é tão transparente quanto possível. Uma interface clara permite que o utilizador não tenha dúvidas e consiga abdicar da utilização de um manual de instruções de funcionamento;
- **Concisão** - A interface diz-se concisa, ou sucinta, quando o utilizador não tem dificuldades em identificar e utilizar os diferentes módulos que a compõe. Para tal a interface deve apenas mostrar a informação mais importante de modo a que a sua utilização não se torne uma actividade entediante e desinteressante para o utilizador. Geralmente, concilia-se clareza com concisão.
- **Familiaridade** - Quando se diz que a interface deve ser familiar, a ideia associada a este conceito é que o seu design apresente semelhanças com algumas interfaces utilizadas frequentemente pelo utilizador em causa. Porém, refira-se o facto de que esta regra não implica que se copie o design de outras interfaces. Por exemplo, se a interface possuir uma barra de navegação, esta deve ser tão semelhante quanto possível às geralmente usadas, pelo utilizador, noutras aplicações. Assim, pretende-se que um utilizador menos experiente comece a utilizar e explorar a interface de um modo simples e rápido.
- **Capacidade de resposta** - Com esta regra pretende-se que a interface seja tão rápida quanto possível a executar as acções e funções pretendidas. Além disso, a interface deve indicar ao utilizador se as suas acções estão a ser bem ou mal sucedidas através da disponibilização de informações objectivas e precisas.

- **Coerência** - A interface deve ser o mais coerente possível, ou seja, deve apresentar conformidade entre factos e ideias, permitindo que os utilizadores possam apreender a informação disponibilizada e aplicá-la, eventualmente, no futuro.
- **Aspecto gráfico** - Embora não seja um factor crucial na implementação de uma interface, convém que esta tenha boa aparência. Quando um utilizador gosta do design da interface que utiliza, a eficácia das suas acções é muito maior.
- **Eficiência** - A eficiência é a regra que sintetiza a famosa expressão “Time is Money”. Assim, pretende-se que a interface apresente um design elegante e cuidado, para além de um número suficiente de atalhos, que permita direccionar grande parte do tempo dispendido no seu manuseamento apenas para a realização das tarefas que motivaram a utilização da interface em causa.
- **Tolerância a erros** - A interface não deve castigar o utilizador por erros cometidos. Pretende-se, então, que as acções indevidas por parte do utilizador possam ser controladas e evitadas. Por exemplo, quando o utilizador pretende eliminar dados de elevada importância deve aparecer uma mensagem que solicite, ao utilizador, a confirmação ou rejeição dessa mesma acção.

No desenvolvimento da interface que motivou a escrita desta dissertação, teve-se em conta as regras definidas anteriormente com o intuito de se alcançar uma interface capaz de apoiar o ensino de semicondutores a nível académico.

5.3 - Implementação com Classes

A programação orientada aos objectos permite definir classes que não são mais do que representações abstractas dos objectos. Se cada objecto possuir características e comportamentos próprios, então uma classe possui também os seus próprios atributos (características dos objectos) e métodos (comportamentos dos objectos) [46].

Os atributos de uma classe representam o estado do objecto, enquanto os métodos representam o comportamento dos mesmos. Se considerarmos o exemplo da definição de uma classe “carro” tem-se:

- **Atributos** - Ligado, Desligado, Velocidade, Marca, Modelo, Número de portas, etc;
- **Métodos** - Ligar, Desligar, Acelerar, Travar, Virar esquerda, etc;

Na definição de uma classe existem ainda os eventos. Estes são responsáveis por alterar o estado e mudar o comportamento dos objectos. Tendo ainda como base o exemplo da classe carro, tem-se:

- **Eventos** - Carregar no pedal do acelerador ou no pedal de travar, virar o volante para a esquerda ou para a direita, rodar a chave da ignição do estado ligado para o estado desligado, etc;

A utilização de classes tem várias vantagens, entre as quais se podem distinguir as seguintes:

- **Simplicidade** - As classes escondem a complexidade do código implementado. O desenvolvimento em classes permite utilizar os diagramas de modelização UML, em particular os de classes, uma vez que permitem explicar o relacionamento entre todas as classes que compõe a aplicação.
- **Reutilização de código** - Uma classe depois de criada pode ser reutilizada por outras aplicações, dado que os elementos da mesma classe partilham o mesmo tipo de atributos e os mesmos métodos.
- **Inclusão dinâmica** - Depois de definida e criada a classe, é possível criar instâncias, ou seja, novos objectos dinâmicos, reduzindo-se, desta forma, o tamanho do projecto final.

Na Figura 5-1 apresenta-se um exemplo de definição da classe carro, supra apresentada. Podem-se definir quatro zonas distintas no código relativo a uma classe, explicadas de seguida:

1. Sempre que se pretender definir uma classe, utiliza-se a palavra “*Type*”. Como se pode ver a classe “*Object*” é a classe mãe da classe “*Carro*”. Por sua vez, o nome da classe fica ao critério do programador.
2. e 3. Nestes pontos aborda-se a questão da visibilidade dos atributos e métodos de uma classe. Por um lado, um atributo do tipo “*Private*” não pode ser lido ou escrito a partir de outra classe. Por outro lado, um método do tipo “*Private*” não pode ser acedido por outra classe. Assim, atributos e métodos deste tipo são exclusivos à classe em causa. Os métodos e atributos do tipo “*Public*” podem ser lidos e escritos (atributos) ou acedidos (métodos) por outra classe.
4. A linguagem “Object Pascal” permite definir funções e procedimentos. As funções permitem receber parâmetros de entrada e retornam valores de saída, sendo definidas recorrendo à palavra “*Function*”. Os procedimentos também recebem parâmetros de entrada mas não retornam valores de saída. Estes definem-se através do código “*Procedure*”. Nesse ponto, deve-se salientar a existência de dois procedimentos especiais, o “*Constructor*” e o “*Destructor*”. O primeiro é responsável por criar e inicializar novas instâncias da classe (novos objectos). O segundo, pelo contrário, é responsável por destruir e eliminar os objectos, permitindo que a memória que estes ocupavam seja libertada.

```

type
  TCarro = class(TObject) 1
  private
    matricula:string; 2

  public
    marca:string;
    modelo:string;
    Proprietario:String; 3
    Estado:string;

    constructor create(); 4
    destructor destroy();
    procedure define_matricula(Marca,Modelo:string);
    function define_estado(estados:string):boolean;

end;

```

Figura 5-1 - Implementação da Classe Carro.

O desenvolvimento da interface utilizando classes motivou um estudo e levantamento de elementos (objectos) que partilhassem o mesmo tipo de atributos e métodos. Desse estudo resultaram algumas classes, tais como UMain, UCircuito, UDíodo, etc, que se explicam de seguida.

Em primeiro lugar, definiu-se que a classe UMain representa as informações que a interface fornece ao utilizador e as acções que lhe possibilita executar. De seguida, e tendo em conta que a interface simula circuitos electrónicos, definiu-se uma classe representativa de todos os circuitos da interface - a classe UCircuito.

Um circuito é composto por vários componentes pelo que seria de esperar que existisse uma classe que os representasse. Embora cada componente tenha algumas propriedades (atributos) próprias, existem outras que são gerais para todos os componentes. Desta forma, implementou-se uma classe que representa os componentes usados e que trata dos atributos gerais destes - UComponentes. No sentido de ter uma classe que tratasse os atributos e procedimentos próprios de cada componente, implementaram-se várias classes que os representassem individualmente. Assim, as classes representativas dos dispositivos usados são as seguintes: UDíodo, UBobine, UCarga, UFontes, UResistência, UFontes_Comando, UTransistor, UReferência.

A propriedade de herança, característica da linguagem “Object Pascal”, e que foi referenciada na secção 4.3, do capítulo 4, permite que uma classe herde atributos e procedimentos de outras. Para tal, é necessário definir uma classe como classe mãe ou super classe e as outras como sub-classes. Assim, definindo-se a classe UComponentes como classe mãe e as classes UDíodo, UBobine, UFontes, UResistência, UFontes_Comando, UTransístor, UReferência como sub-classes, pretende-se que as sub-classes possam herdar os atributos e procedimentos genéricos, facilitando o desenvolvimento do código. Esta propriedade será explicada com detalhe nas secções seguintes.

Ao nível da apresentação dos resultados utilizou-se uma classe para os gráficos 3D - Ugraficos 3D, uma para os gráficos 2D - Ugráficos 2D e outra que permitisse ao utilizador escolher os gráficos que pretende visualizar - UEscolha Gráficos. Estas classes assumem elevada importância porque possibilitam a visualização dos gráficos do Dóido e do IGBT ao mesmo tempo.

De forma a permitir que o utilizador possa alterar os valores dos componentes constituintes de um circuito, definiram-se três classes (UPainel, UPainel_díodos, UPainel_Transístor). Estas classes são constituídas por botões e janelas de edição permitindo definir os vários valores dos componentes.

Ao nível das explicações teóricas, definiu-se uma classe (UExplicacao_teorica) com o intuito de fornecer ao utilizador, quer a explicação de funcionamento do circuito, quer a explicação das formas de onda.

Com o intuito de melhorar e potenciar a visualização dos gráficos definiram-se duas classes, UMarcadores e UZoom. A primeira permite utilizar uma característica dos gráficos a duas dimensões para medir intervalos de tempo, ou de valores. A segunda permite efectuar vários tipos de zoom aos gráficos, melhorando a interpretação destes.

De salientar ainda que foi implementada uma classe responsável apenas pelo tratamento de erros ocorridos durante a execução da aplicação - a classe UErro.

5.4 - Definição das classes na interface

1. Classe UComponentes

A classe UComponentes - Figura 5-2 - é constituída por atributos (Figura 5-2-2) e métodos genéricos (Figura 5-2-3), que podem ser acedidos e utilizados por outras classes, como já foi referido anteriormente, na secção 5.3 - .

UComponentes		1
-Etiqueta : string		2
-Legenda_etiqueta : string		
-x1 : int		
-x2 : int		
-y1 : int		
-y2 : int		3
+Create()		
+Destroy()		
+Define_posicao(in posx1 : int, in posx2 : int, in posy1 : int, in posy2 : int)		
+Define_legenda(in legenda : string)		
+Define_nome(in nome : string)		

Figura 5-2 - Classe UComponentes 1- Nome da Classe; 2- Atributos; 3- Métodos.

2. Classe UDíodo

Na Figura 5-3 apresenta-se a classe UDíodo. Esta classe representa o conjunto de díodos que têm os mesmos atributos (Figura 5-32-2) e os mesmos métodos (Figura 5-3-3).

UDíodo		1
-a : string		2
-wb : string		
-tau : string		
-h : string		
-nb : string		
-valores : string		3
+Create()		
+Destroy()		
+Define_parametros(in area : string, in larg_base : string, in n_dop_base : string, in temp_m_vida : string, in par_rec : string)		
+Cria_ficheiros_valores(in caminho : string)		

Figura 5-3 - Classe UDíodo. 1-Nome da Classe; 2- Atributos; 3- Métodos

3. Classe UBobine

A classe UBobine é constituída pelas indutâncias utilizadas nos circuitos implementados. Nas partes 2 e 3 da Figura 5-4 apresentam-se, respectivamente, os atributos e os métodos que caracterizam esta classe.

UBobine	1
-Valor1 : string -Valor2 : string -Valor3 : string -Valor_simular : string -valores : string	2
+Create() +Destroy() +Define_valores(in Val1 : string, in val2 : string, in val3 : string) +Cria_ficheiros_valores(in caminho : string)	3

Figura 5-4 - Classe UBobine. 1- Nome, 2- Atributos, 3-Métodos.

4. Classe UFontes

A classe UFontes Figura 5-5 corresponde às fontes de tensão utilizadas nos circuitos implementados. Os atributos desta classe foram definidos tendo em conta que as fontes de tensão utilizadas são de tensão contínua. No entanto, e caso seja necessário adicionar uma fonte de tensão alternada, no futuro, basta adicionar novos atributos à classe em questão. Esta propriedade revela mais uma vez a importância da utilização de classes no desenvolvimento do código da aplicação.

UFontes	1
-Valor1 : string -Valor2 : string -Valor3 : string -Valor_simular : string -valores : string	2
+Create() +Destroy() +Define_valores(in Val1 : string, in val2 : string, in val3 : string) +Cria_ficheiros_valores(in caminho : string)	3

Figura 5-5 - UFontes. 1- Nome, 2- Atributos, 3- Métodos.

5. Classe UResistência

A classe UResistência (Figura 5-6) é representativa das resistências utilizadas nos circuitos implementados. Nas partes 2 e 3, da Figura 5-6, são expostos, respectivamente, os atributos e métodos desta classe.

UResistência	1
-Valor1 : string -Valor2 : string -Valor3 : string -Valor_simular : string -valores : string	2
+Create() +Destroy() +Define_valores(in Val1 : string, in val2 : string, in val3 : string) +Cria_ficheiros_valores(in caminho : string)	3

Figura 5-6 - Classe UResistência. 1- Nome, 2- Atributos, 3- Métodos.

6. Classe UFontes_Comando

A classe UFontes_Comando Figura 5-7 - é representativa da fonte de corrente que se estabelece no conjunto L1 e LF (Capítulo 3), quando o transistor MOS entra em condução, do comportamento da bobine (fonte de corrente constante) durante o tempo de comutação do mesmo e, por último, da tensão de comando utilizada no circuito de teste de um transistor IGBT. Estas fontes partilham os mesmos atributos (Figura 5-7-2) e métodos (Figura 5-7-3).

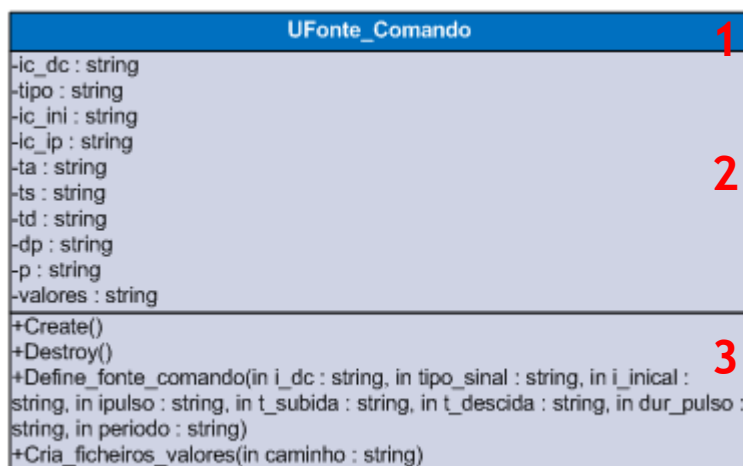


Figura 5-7 - Classe UFontes_Comando. 1- Nome, 2- Atributos, 3- Métodos.

7. Classe UTransistor

Esta classe representa os transistores de junção bipolar utilizados nos circuitos implementados. Estes transistores partilham os mesmos atributos e métodos, os quais estão apresentados na Figura 5-8.

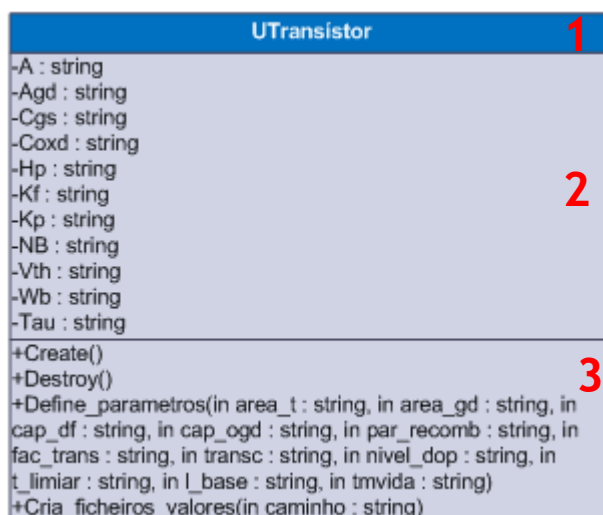


Figura 5-8 - Classe UTransistor. 1- Nome, 2- Atributos, 3- Métodos.

8. Classe UReferência

A classe UReferência - Figura 5-9 - representa, como o próprio nome indica, as referências assinaladas nos circuitos implementados. Nestes mesmos circuitos, esta classe foi utilizada para disponibilizar o valor do potencial da Terra (*Ground*), que, por convenção, é zero Volt. No entanto, e tendo em conta os seus atributos (Figura 5-9-2) e métodos (Figura 5-9-3), podia, por exemplo, corresponder a um valor de referência de tensão num determinado circuito.

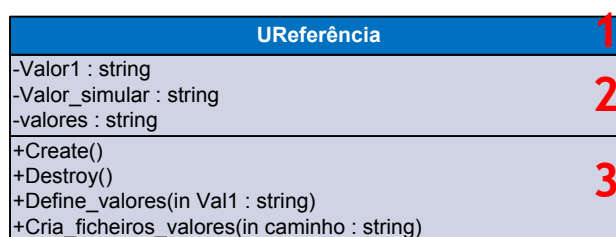


Figura 5-9 - Classe UReferência. 1- Nome, 2- Atributos, 3- Métodos.

9. Classe UCircuito

Esta classe disponibiliza os vários circuitos que podem ser simulados na interface desenvolvida. Assim, é possível, por exemplo, definir dois circuitos (instâncias desta classe) que representam os circuitos de teste do diodo e do transistor (IGBT). Os atributos e os métodos desta classe estão apresentados na Figura 5-10. Deve-se notar que na classe UCircuito são criadas as instâncias dos componentes a utilizar nos circuitos (Figura 5-10-2a). Como se pode verificar, existem várias bobinas, por exemplo as bobinas LF e L1, que são

instâncias da classe UBobine, reforçando-se, assim, a ideia de que a criação de classes de objectos com os mesmos atributos e métodos teve uma importância relevante no desenvolvimento do presente trabalho.

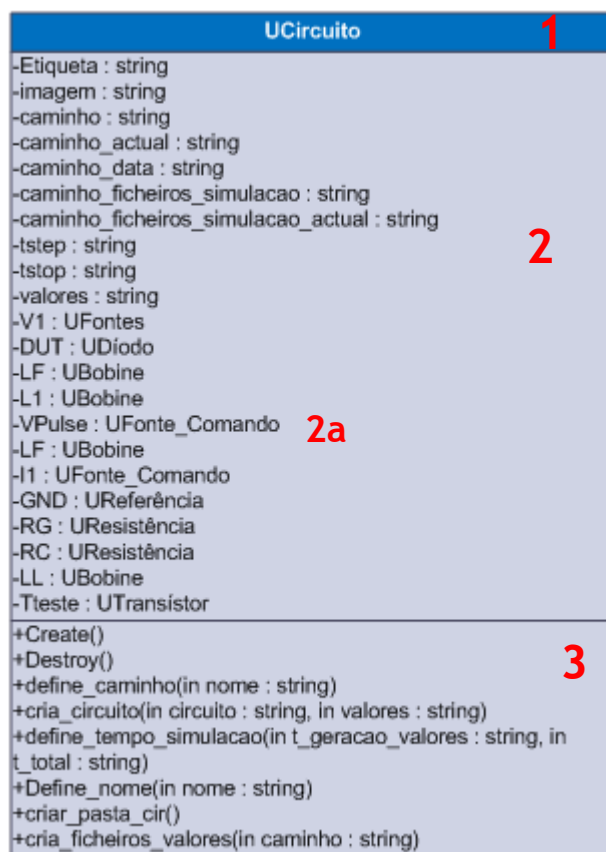


Figura 5-10 - Classe UCircuito. 1- Nome, 2- Atributos, 2a- Instâncias dos vários componentes, 3- Métodos.

10. Classe UGraficos_2D

A classe UGraficos_2D representa os gráficos a duas dimensões - Figura 5-11. Esta classe é desenvolvida fazendo uso de uma “Form” onde é possível não só visualizar as formas de onda da tensão e da corrente em função do tempo de simulação mas também calcular os valores máximos e mínimos de tensão e corrente, bem como os tempos em que estes ocorrem. Tendo em conta que esta classe é construída com o recurso a uma “Form”, alguns dos seus atributos e métodos representam os botões, os painéis, as caixas de texto e os eventos associados a cada um destes atributos, estando, desta forma, agrupados em “Atributos gráficos”⁴ e “Métodos gráficos”² - (Figura 5-11-2 e Figura 5-11-5). Existem ainda outros atributos e

⁴ - Optou-se por não representar todos os métodos e atributos associados aos botões, caixas de texto, etc. Isto deve-se ao facto destes serem componentes gráficos e os seus métodos serem apenas utilizados para navegação entre páginas, accionar caixas de texto, entre outras acções. Os métodos e atributos mais importantes são apresentados em “Atributos” e “Métodos”.

métodos que foram definidos e que estão apresentados respectivamente na Figura 5-11-3 e na Figura 5-11-4.

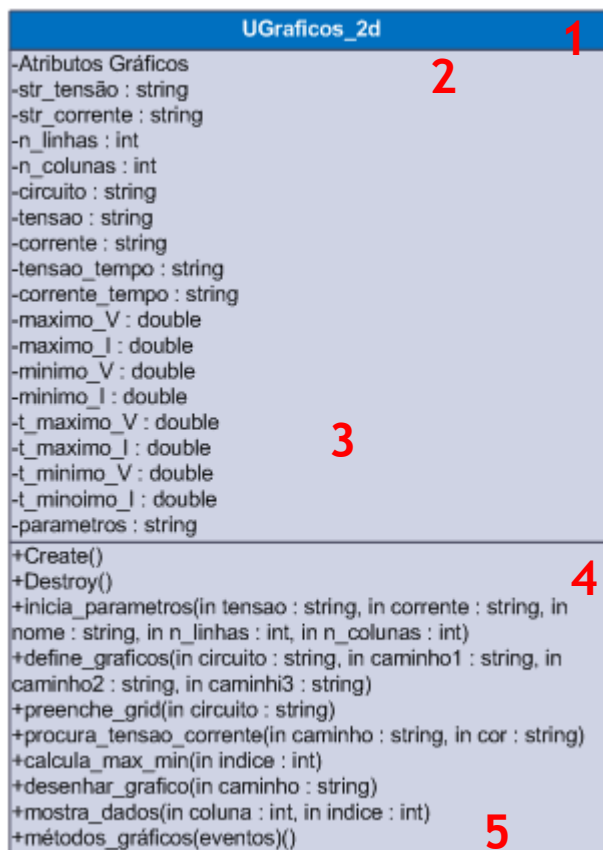


Figura 5-11 - Classe UGraficos_2d. 1- Nome, 2- Atributos gráficos ("Form"), 3- Atributos, 4- Métodos, 5- Métodos gráficos (eventos).

11. Classe Ugraficos_3D

A classe Ugraficos_3D é apresentada na Figura 5-12. Esta classe é responsável por apresentar os gráficos a três dimensões e, tal como a classe UGraficos_2D, é implementada fazendo recurso a uma "Form". Esta é composta por vários, botões, caixas de texto, "radio buttons", gráficos, tabelas etc. Desta forma, existem atributos e métodos associados a estes elementos gráficos que são agrupados respectivamente em "atributos gráficos" (Figura 5-12-2) e "elementos gráficos" (Figura 5-12-5). Para além destes, definiram-se também outros atributos e métodos, os quais são apresentados nas partes 3 e 4 da Figura 5-12.

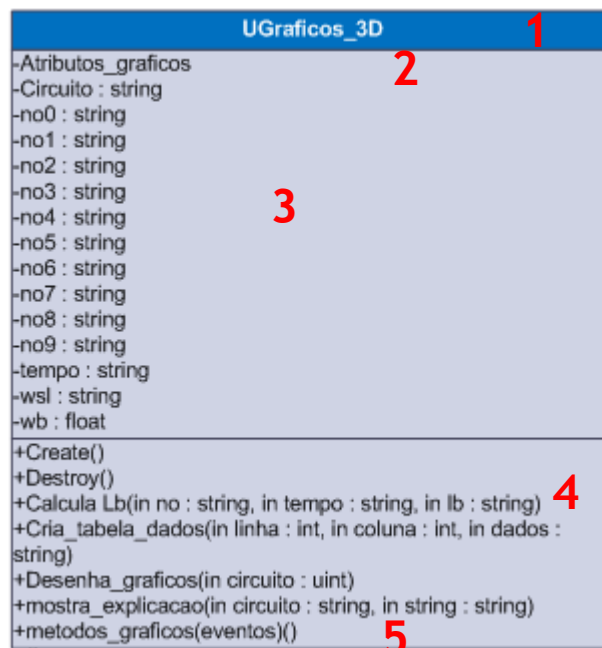


Figura 5-12 - Classe Ugraficos_3D. 1-Nome, 2-Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos Gráficos.

12. Classe UEscolha_Graficos

A classe UEscolha_graficos permite escolher os gráficos que se pretende desenhar. Esta classe é construída tendo como base uma “Form” e, tal como nas classes UGraficos_2D e UGraficos_3D, é composta por vários elementos gráficos com atributos e propriedades que serão agrupados em “atributos gráficos” e “métodos gráficos”. A implementação desta classe permite que se criem tantas instâncias quanto as necessárias, evitando desta forma repetição de código.

Esta classe foi pensada e implementada com o intuito de colocar à disposição do utilizador um menu simples e organizado, com acesso às simulações efectuadas e guardadas em histórico.

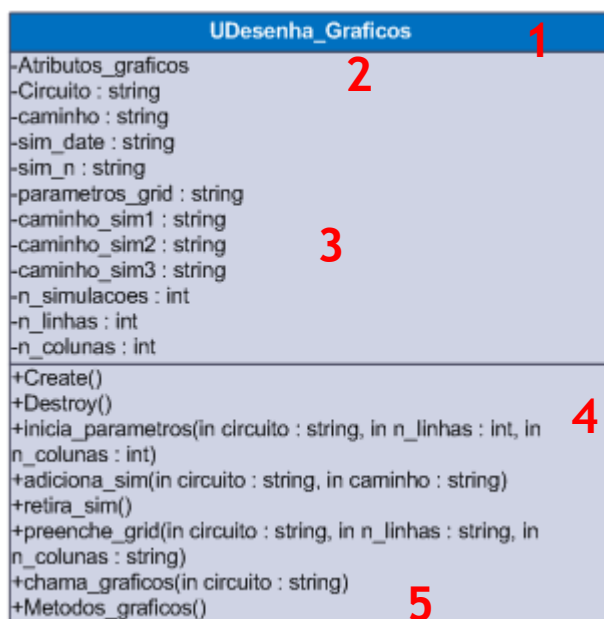


Figura 5-13 - Classe UEscolha_graficos. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos Gráficos.

13. Classe UExplicacao_teorica

Na Figura 5-14 apresenta-se a classe UExplicação_Teórica. Esta classe é responsável por mostrar as diversas explicações teóricas sobre o funcionamento dos circuitos e das simulações guardadas em histórico. Esta classe utiliza uma “Form” e permite ao utilizador inserir correcções, apagar, ou simplesmente ler as informações facultadas. A utilidade desta classe passa por possibilitar a criação simultânea de várias janelas de explicações teóricas, no mesmo gráfico, ajudando o utilizador a compreender, com maior facilidade, as zonas características das formas de onda de tensão e corrente, em simulações típicas do comportamento dos semicondutores utilizados.

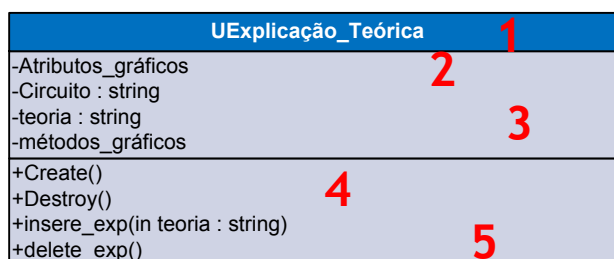


Figura 5-14 - Classe UExplicação_teorica. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.

14. Classe UPainel

A classe UPainel permite que o utilizador possa escolher ou alterar o valor de um determinado componente para, posteriormente, realizar a simulação. Esta classe representa os painéis dos componentes que pertencem as seguintes classes: UFontes, UBobine, UResistência. Tendo em conta que não se pretende que este painel apareça ao mesmo tempo em diferentes componentes, não foi implementado um painel para cada componente isolado. Assim, e como os elementos das classes UFontes, UBobine e UResistência partilham os mesmos atributos e métodos, implementou-se um painel característico destas três classes.

Este painel construiu-se com base numa “Form” e, como tal, os atributos e métodos gráficos não serão detalhados. No entanto, definiram-se atributos e métodos que são apresentados na Figura 5-15.

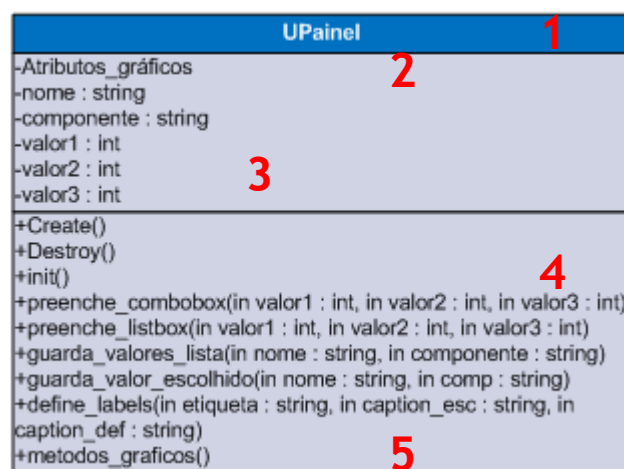


Figura 5-15 - Classe UPainel. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.

15. Classe UPainel_Díodos

Esta classe, tal como o nome indica, disponibiliza um painel de escolha e edição de valores para um componente em concreto - o díodo. Assim, o utilizador é capaz de alterar e seleccionar as características do díodo a utilizar na simulação de um circuito. A classe UPainel_Díodos utiliza uma “Form”, sendo composta por botões e caixas de texto, onde o utilizador pode alterar os parâmetros do díodo de uma forma fácil e eficiente.

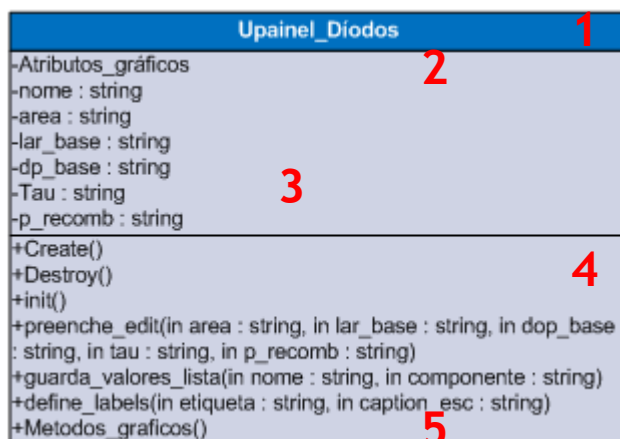


Figura 5-16 - Classe UPainel_Díodos. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.

16. Classe UPainel_Transístor

A classe UPainel_Transistor corresponde ao painel que permite ao utilizador alterar os parâmetros de cada transístor presente nos circuitos. Esta classe também utiliza uma “Form” e é constituída por diversos elementos gráficos, aos quais estão associados vários atributos e métodos - Atributos e Métodos gráficos. A Figura 5-17 apresenta a classe Upainel_Transístor.

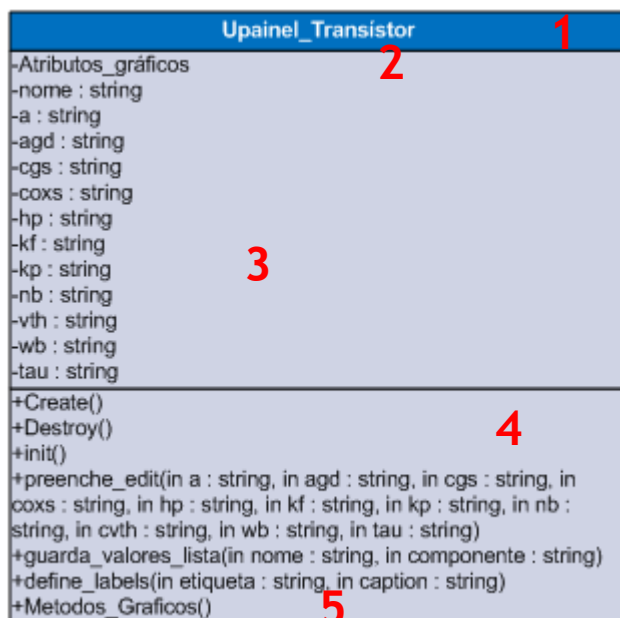


Figura 5-17 - Classe Upainel_transistor. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.

17. Classe UPainel_Fontes_Comando

A classe apresentada na Figura 5-18 - UPainel_Fontes_Comando - permite ao utilizador seleccionar as características da fonte de comando. Esta classe é constituída pela fonte de corrente constante no díodo (Capítulo 3), a fonte de corrente de comando no circuito do díodo e a fonte de tensão de comando no circuito do IGBT. Tal como acontece com as anteriores, esta classe é construída utilizando uma “Form”, com diversos atributos e métodos que estão associados aos elementos gráficos (Atributos_graficos e Métodos_graficos).

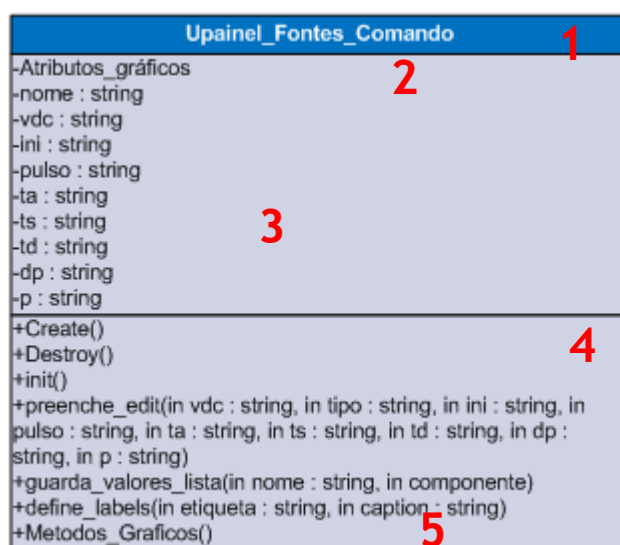


Figura 5-18 - Classe Upainel_fontes_comando. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.

18. Classe Upainel_Marcadores

A classe Upainel_Marcadores, Figura 5-19, foi criada com o intuito de disponibilizar ao utilizador a capacidade de colocar marcadores nos gráficos bidimensionais da tensão e da corrente, permitindo, desta forma, a medição de tempos ou de grandezas eléctricas. Esta classe foi também construída com uma “Form” e, como tal, apresenta atributos e métodos gráficos que são apresentados em anexo.

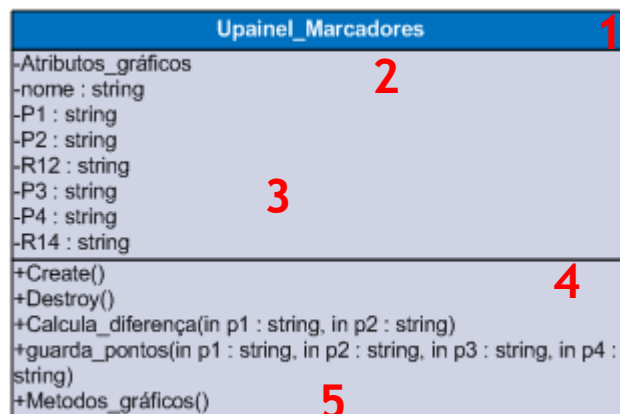


Figura 5-19 - Classe Upainel_Marcadores. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.

19. Classe UPainel_Zoom

De forma a permitir que o utilizador tenha liberdade de visualizar com detalhe os gráficos bidimensionais, desenvolveu-se a classe Upainel_Zoom - Figura 5-20. Esta classe utiliza uma “Form” e permite fazer vários tipos de zoom em circuitos independentes (ampliar, diminuir, arrastar horizontalmente e verticalmente). Assim, podem-se criar duas instâncias desta classe, associando uma ao gráfico da tensão e outra ao gráfico da corrente, podendo-se, assim, ter acção independente sobre cada gráfico.

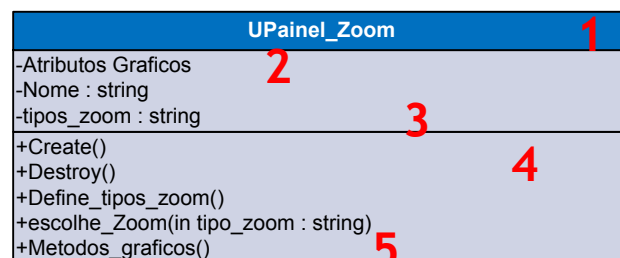


Figura 5-20 - Classe UPainel_Zoom. . 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.

20. Classe UErro

A classe UErro - Figura 5-21 - é extremamente útil no desenvolvimento da interface. Com a sua utilização é possível definir tipos de erros que ocorrem em diversas situações, mas cuja solução é a mesma, evitando-se a repetição de código. Esta classe também pode ser usada para mostrar informação sobre as acções do utilizador, através de mensagens concretas e sucintas, como por exemplo, “Simulação efectuada com sucesso”.

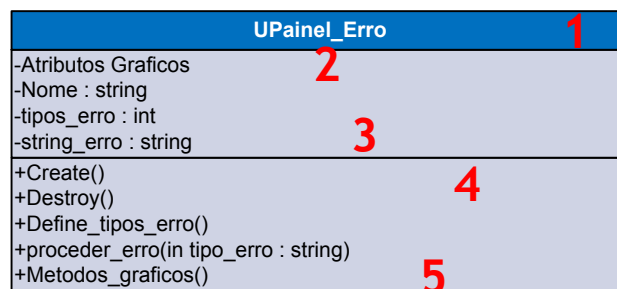


Figura 5-21 - Classe UErro. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.

21. Classe UMain

A classe UMain - Figura 5-22- é a classe principal da interface, sendo responsável por fornecer todas as informações e por assegurar que todas as acções podem ser executadas pelo utilizador. Esta classe é construída com base numa “Form”, definida como a “Form” principal. Desta forma, todas as “Forms” que possam surgir por acção do utilizador são secundárias. Ao fechar a “Form Main”, o programa é encerrado.

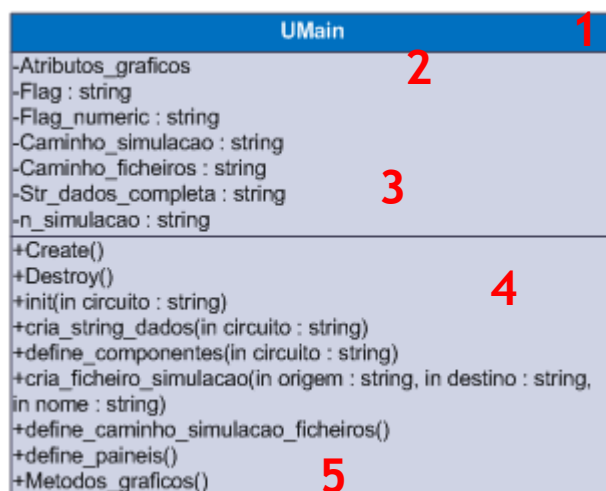


Figura 5-22 - Classe UMain. 1- Nome, 2- Atributos Gráficos, 3- Atributos, 4- Métodos, 5- Métodos gráficos.

22. Unit funções

Depois de apresentadas todas as classes que compõem a interface gráfica, convém fazer referência a uma “Unit” onde são definidas várias funções que não pertencem a nenhuma classe desenvolvida - Figura 5-23. As seguintes funções fazem parte dessa “unit”:

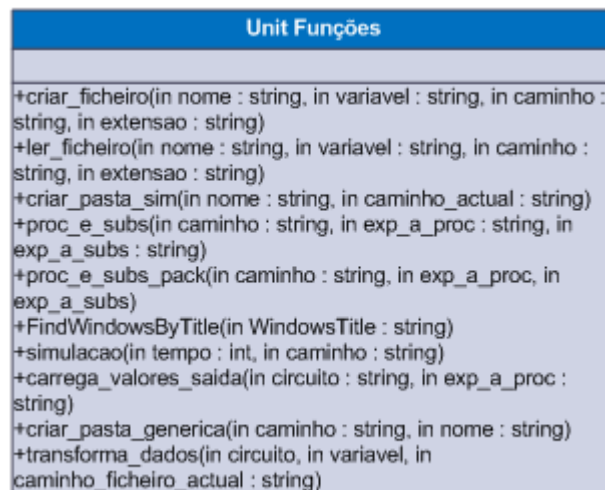


Figura 5-23 - Unit Funções.

Na secção 5.27 deste capítulo serão apresentadas não só algumas funções das classes implementadas, mas também algumas das funções presentes na “Unit Funções” e que se revelaram imprescindíveis para completar os objectivos propostos para este trabalho.

5.5 - Diagramas UML

UML é uma linguagem de modelização e não uma metodologia de desenvolvimento, ou seja, permite que através de diagramas estáticos e dinâmicos se perceba a utilidade dos objectos e o seu relacionamento. Existem vários diagramas UML que permitem explicar o funcionamento e comportamento dos objectos. Nesta dissertação são implementados dois diagramas UML [10]: Classes e Casos de Uso. Estes diagramas permitem explicar o funcionamento da interface desenvolvida.

5.5.1 - UML Diagrama de Classes

Os diagramas de classes mostram as diferentes classes que fazem parte de um sistema, a forma como estas se relacionam, e os métodos e atributos que as caracterizam. Os relacionamentos estáticos entre as diversas classes dizem respeito às classes que se “conhecem” ou as classes que “fazem parte” de outras. No entanto, estes diagramas não mostram a troca de mensagem entre as diversas classes, pelo que são denominados de diagramas estáticos.

Tendo em conta a apresentação individual das classes realizada neste capítulo e de forma a ser possível visualizar os relacionamentos entre as diversas classes, optou-se por apenas representar a classe com o seu nome, diminuindo substancialmente o tamanho do diagrama de classes -Figura 5-24.

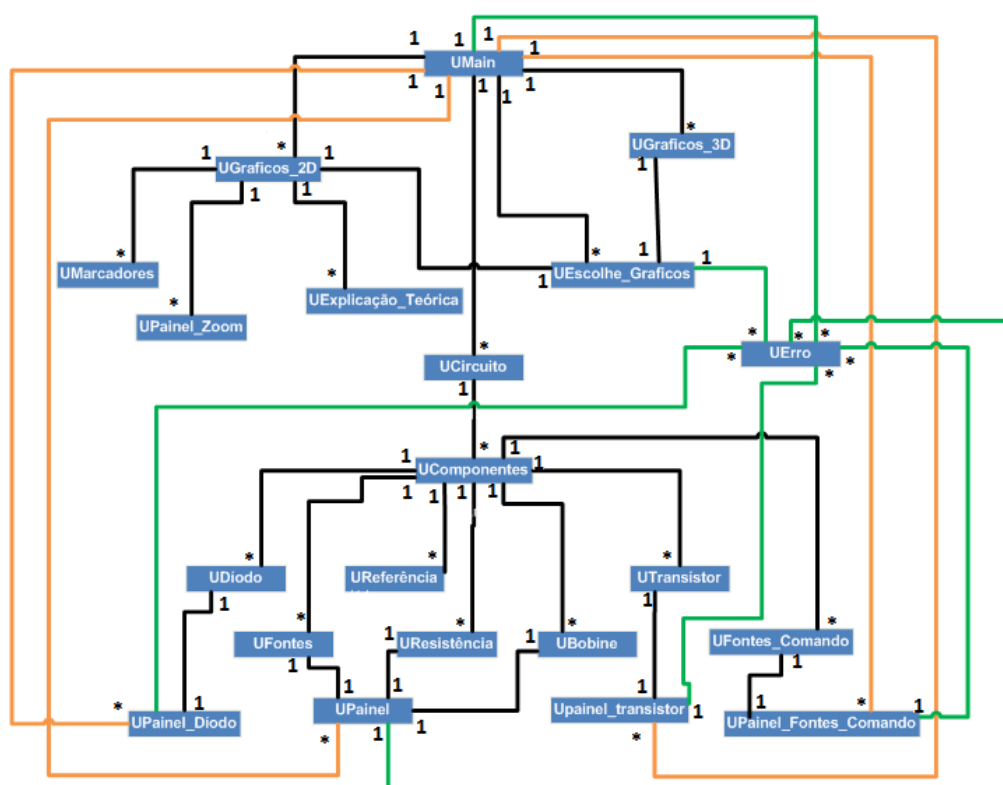


Figura 5-24 - Diagrama UML de Classes da interface desenvolvida.

5.5.2 - UML Casos de Uso

Os diagramas UML Casos de Uso são descrições das interações típicas entre os utilizadores do sistema e o sistema propriamente dito, ou seja, representam a interface externa do sistema e especificam um conjunto de exigências que o sistema deve fazer. Assim, um caso de uso refere-se a um objectivo concreto que o utilizador pretende alcançar quando interage com a interface.

Estes diagramas são compostos por “actores” que representam as entidades externas ao sistema que interagem com este, participando num caso de uso. Na Figura 5-25, apresenta-se o diagrama de casos de uso genérico da interface desenvolvida.

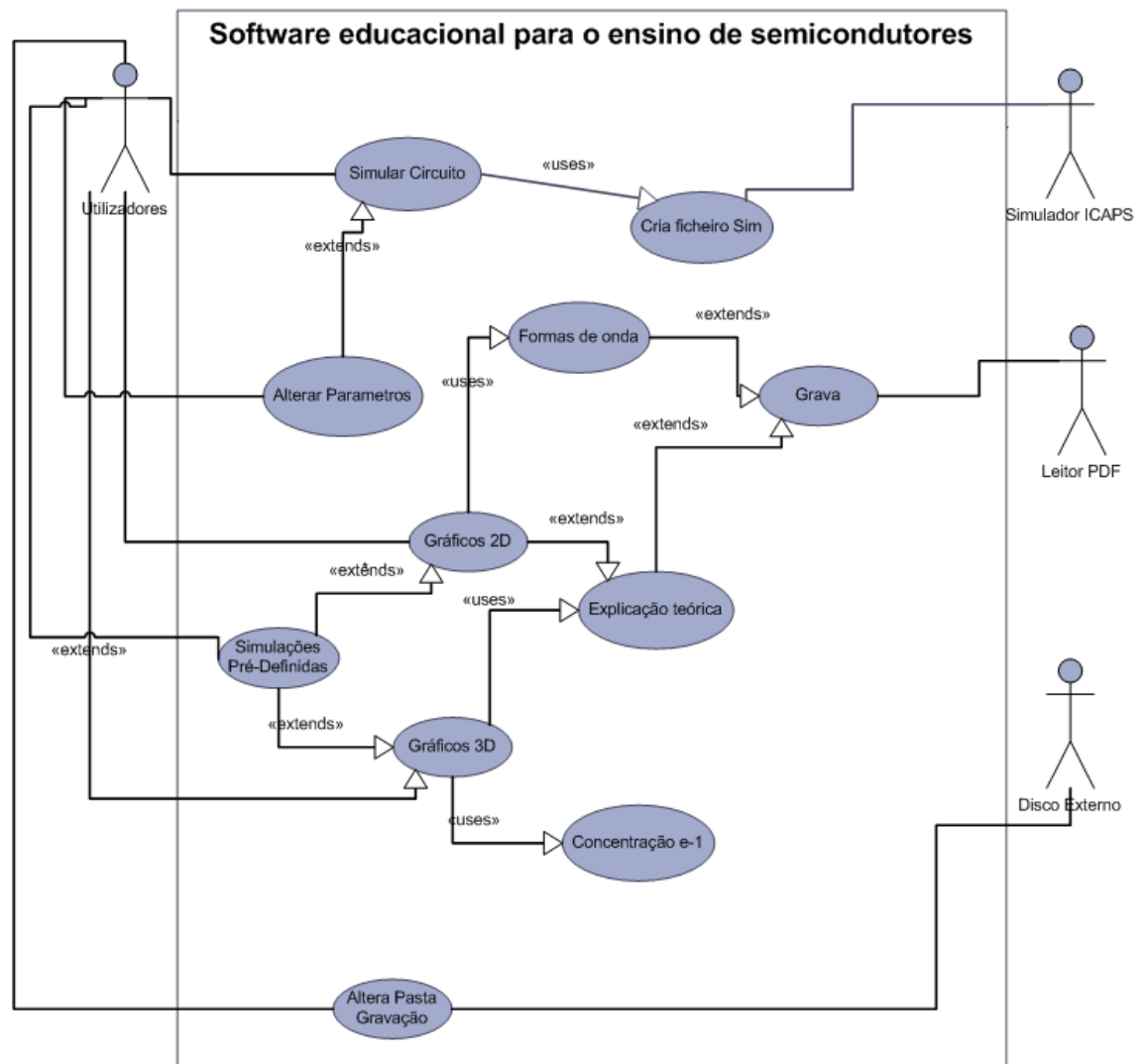


Figura 5-25 - Diagrama UML Casos de Uso.

Cada caso de uso apresentado tem uma descrição específica e desta forma são explicados de seguida para três casos de uso:

- Simular Circuito - Tabela 5-1.
- Simulações Pré-Definidas - Tabela 5-2.
- Alterar pasta gravação - Tabela 5-3.
- Gravar no formato .pdf - Tabela 5-4.

Tabela 5-1 - Descrição do caso de uso “Simular Circuito”.

Condicionantes \ Casos de Uso	Simular Circuito
Quem Inicia	Utilizador
Objectivo	Simular Circuito
Descrição	<ul style="list-style-type: none"> - Abre interface de simulação - Escolhe o circuito que pretende simular - Carrega no botão “simular” - Interface cria ficheiro de simulação com os parâmetros dos componentes definidos, - Simula o circuito recorrendo a um actor do sistema (ICAPS) que produz os resultados de saída.
Outros Intervenientes	Actor do sistema - ICAPS
Resultado	Ficheiro Simulado, Dados de saída extraídos, Pronto para visualizar gráficos.

Tabela 5-2 - Descrição do caso de uso Simulações Pré-Definidas.

Condicionantes \ Casos de Uso	Simulações Pré-Definidas
Quem Inicia	Utilizador
Objectivo	Visualizar gráficos de um simulação pré-definida
Descrição	<ul style="list-style-type: none"> - Abre interface de simulação - Utilizador Carrega “botão gráficos” - Interface abre “Form_Escolhe_gráficos” - Utilizador escolhe a simulação que pretende - Interface carrega os dados referentes a essa simulação - Utilizador carrega “botão gráficos 2D” ou “gráficos 3D” - Interface disponibiliza os gráficos e as explicações teóricas sobre os mesmos - O utilizador pode guardar essas informações num ficheiro “.pdf”

Outros Intervenientes	Actor do sistema - Leitor PDF
Resultado	Visualização dos Gráficos e das explicações teóricas associadas.

Tabela 5-3 - Descrição do caso de uso Alterar Pasta Gravação.

Condicionantes \ Casos de Uso	Alterar Pasta Gravação
Quem Inicia	Utilizador
Objectivo	Alterar caminhos de gravação
Descrição	<ul style="list-style-type: none"> - Abre interface - Utilizador carrega botão “Alterar Caminhos” - Interface disponibiliza “Form_altera_caminhos” - Utilizador altera os caminhos que pretender, por exemplo, pode gravar no seu disco pessoal de armazenamento interno - Interface associa aos caminhos pré-definidos, o caminho actual definido pelo utilizador.
Outros Intervenientes	
Resultado	Caminhos alterados com sucesso

Tabela 5-4 - Descrição do caso de uso Gravar no formato .pdf .

Condicionantes \ Casos de Uso	Gravar no formato .pdf
Quem Inicia	Utilizador
Objectivo	Exportar informação para pdf
Descrição	<ul style="list-style-type: none"> - Abre interface de simulação - Utilizador Carrega “botão gráficos” - Interface abre “Form_Escolhe_gráficos” - Utilizador escolhe a simulação que pretende - Interface carrega os dados referentes a essa simulação. - Utilizador carrega botão “Grava PDF”. - Interface abre uma janela onde pede o nome

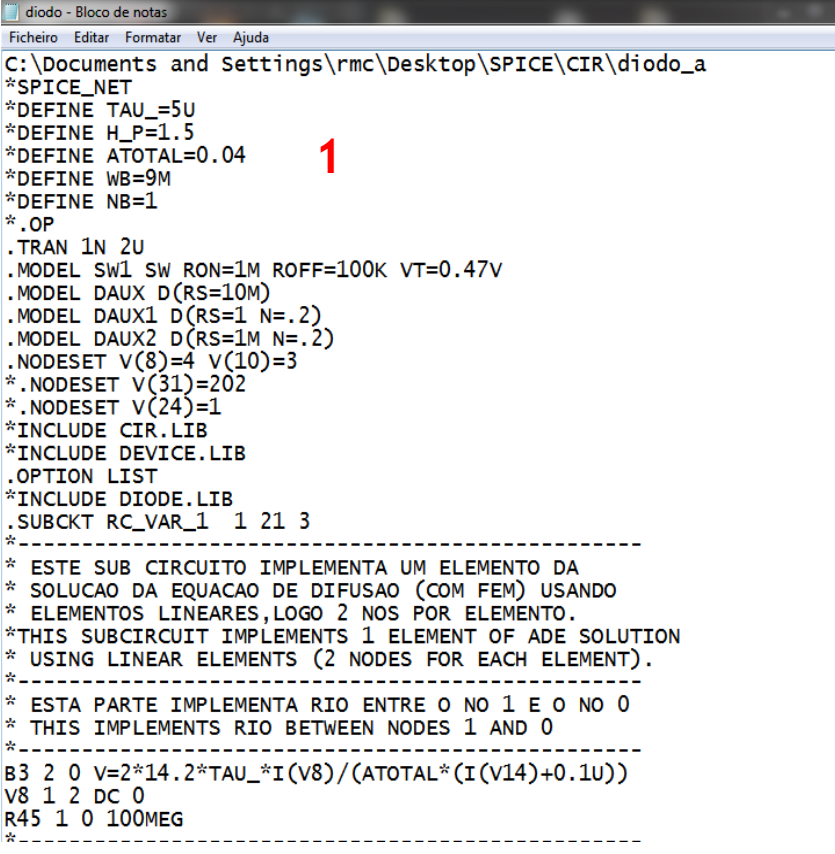
	do ficheiro a guardar. - Interface abre o ficheiro resultante.
Outros Intervenientes	Leitor de PDF.
Resultado	Ficheiro criado com sucesso

5.6 - Funções importantes

Nesta secção são apresentadas algumas funções que se revelaram de extrema importância para a concretização dos objectivos propostos. As funções apresentadas são relativas ao circuito do diodo, no entanto funcionam de igual modo para o circuito do IGBT.

Para se simular um circuito através do módulo IsSpice do programa ICAPS, é necessário fornecer-lhe um ficheiro com a extensão “.ckt” onde está guardada toda a informação do circuito a simular.

Na Figura 5-26 apresenta-se um extracto do ficheiro utilizado para a simulação do circuito do diodo.



```

diode - Bloco de notas
Ficheiro  Editar  Formatar  Ver  Ajuda
C:\Documents and Settings\rmc\Desktop\SPICE\CIR\diode_a
*SPICE_NET
*DEFINE TAU_=5U
*DEFINE H_P=1.5
*DEFINE ATOTAL=0.04
*DEFINE WB=9M
*DEFINE NB=1
*.OP
.TRAN 1N 2U
.MODEL SW1 SW RON=1M ROFF=100K VT=0.47V
.MODEL DAUX1 D(RS=10M)
.MODEL DAUX2 D(RS=1M N=.2)
.NODESET V(8)=4 V(10)=3
*.NODESET V(31)=202
*.NODESET V(24)=1
*INCLUDE CIR.LIB
*INCLUDE DEVICE.LIB
.OPTION LIST
*INCLUDE DIODE.LIB
.SUBCKT RC_VAR_1 1 21 3
*-----
* ESTE SUB CIRCUITO IMPLEMENTA UM ELEMENTO DA
* SOLUCAO DA EQUACAO DE DIFUSAO (COM FEM) USANDO
* ELEMENTOS LINEARES, LOGO 2 NOS POR ELEMENTO.
* THIS SUBCIRCUIT IMPLEMENTS 1 ELEMENT OF ADE SOLUTION
* USING LINEAR ELEMENTS (2 NODES FOR EACH ELEMENT).
*-----
* ESTA PARTE IMPLEMENTA RIO ENTRE O NO 1 E O NO 0
* THIS IMPLEMENTS RIO BETWEEN NODES 1 AND 0
*-----
B3 2 0 V=2*14.2*TAU_*I(V8)/(ATOTAL*(I(V14)+0.1U))
V8 1 2 DC 0
R45 1 0 100MEG
*-----

```

Figura 5-26 - Trecho de código do ficheiro Diodo.ckt. 1- Parâmetros do Díodo.

Como é possível verificar, no ponto 1 da Figura 5-26, aparecem os parâmetros do díodo. De forma a substituir as variáveis pelos parâmetros definidos pelo utilizador implementou-se uma função que procurasse os parâmetros e procedesse à sua substituição. O fluxograma de funcionamento da função é apresentado na Figura 5-27.

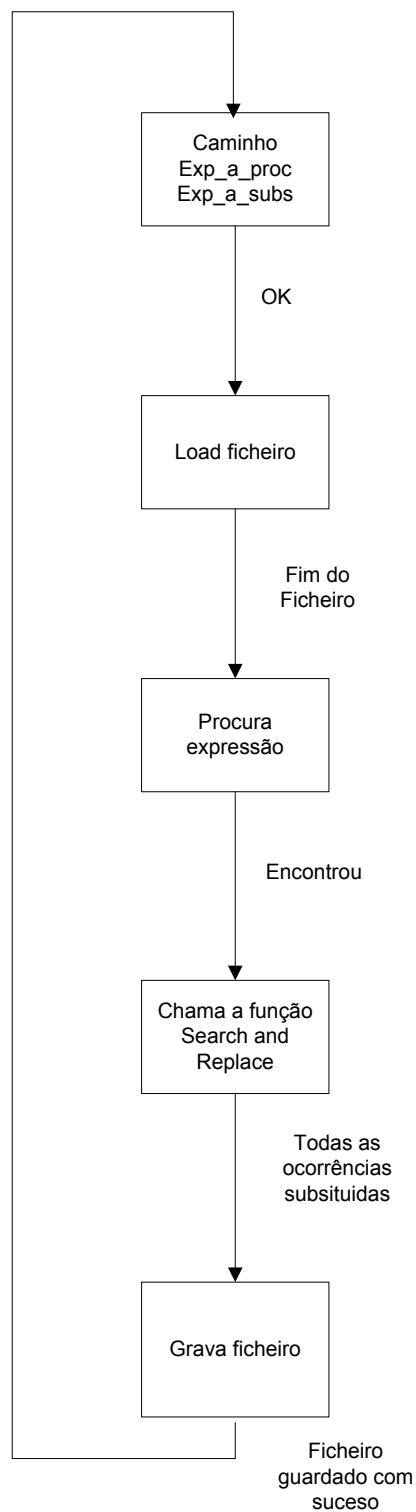
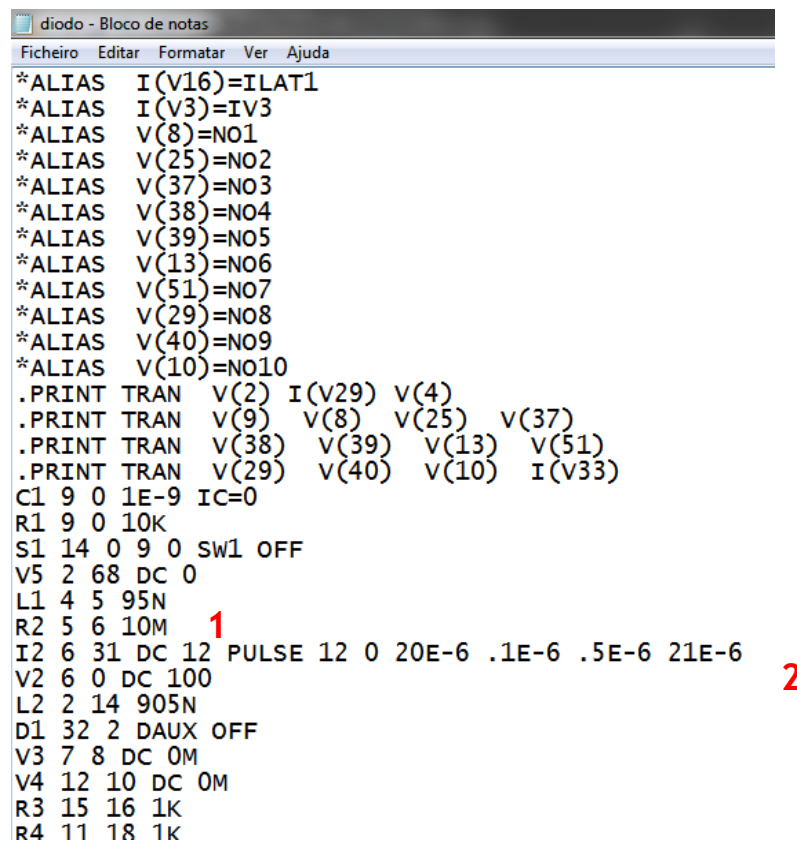


Figura 5-27 - Fluxograma da função “Proc_e_subs”.

Analisando novamente o ficheiro “diodo.ckt”, existem mais variáveis para serem substituídas, nomeadamente as fontes de corrente e as bobines, como é apresentado nos pontos 1 e 2 da Figura 5-28.



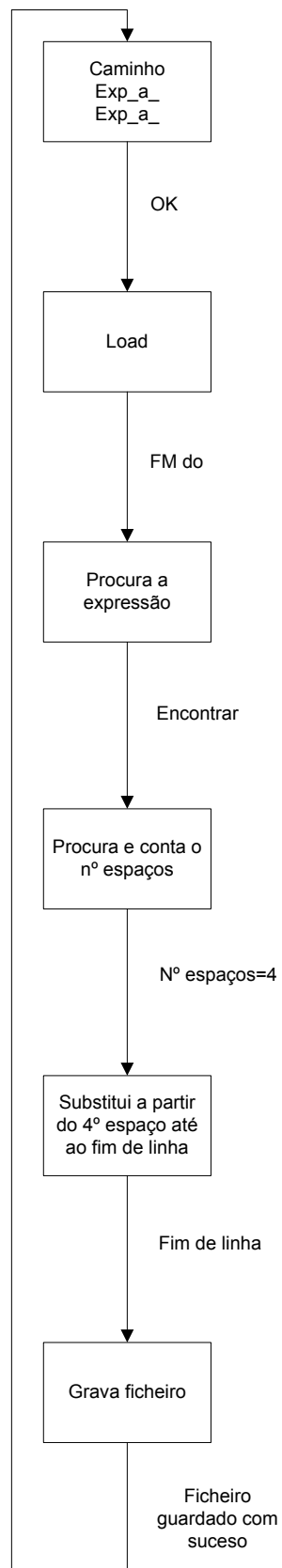
```

diodo - Bloco de notas
Ficheiro Editar Formatar Ver Ajuda
*ALIAS I(V16)=ILAT1
*ALIAS I(V3)=IV3
*ALIAS V(8)=NO1
*ALIAS V(25)=NO2
*ALIAS V(37)=NO3
*ALIAS V(38)=NO4
*ALIAS V(39)=NO5
*ALIAS V(13)=NO6
*ALIAS V(51)=NO7
*ALIAS V(29)=NO8
*ALIAS V(40)=NO9
*ALIAS V(10)=NO10
.PRINT TRAN V(2) I(V29) V(4)
.PRINT TRAN V(9) V(8) V(25) V(37)
.PRINT TRAN V(38) V(39) V(13) V(51)
.PRINT TRAN V(29) V(40) V(10) I(V33)
C1 9 0 1E-9 IC=0
R1 9 0 10K
S1 14 0 9 0 SW1 OFF
V5 2 68 DC 0
L1 4 5 95N
R2 5 6 10M 1
I2 6 31 DC 12 PULSE 12 0 20E-6 .1E-6 .5E-6 21E-6 2
V2 6 0 DC 100
L2 2 14 905N
D1 32 2 DAUX OFF
V3 7 8 DC 0M
V4 12 10 DC 0M
R3 15 16 1K
R4 11 18 1K

```

Figura 5-28- Trecho do ficheiro “Diodo.ckt”. 1- Bobine, 2- Fonte de Corrente

Pela análise da Figura 5-28, verifica-se que a substituição destes parâmetros não é igual à substituição dos parâmetros do diodo, pelo que foi necessário implementar uma nova função de pesquisa que substituisse uma determinada expressão por outra quando encontrasse o componente em questão. A técnica utilizada nesta função passa por contar o número de espaços em branco que existem desde o nome do componente, por exemplo “L1” até ao valor que é pretendido substituir, para posteriormente se substituir pelo valor pretendido. O fluxograma da função implementada é apresentado na Figura 5-29.

Figura 5-29 - Fluxograma da função `proc_e_subs_pack`.

Com o ficheiro de simulação definido correctamente, tem de existir uma função que proceda à simulação do mesmo. A Figura 5-30 apresenta fluxograma de funcionamento da função “Simulação”.

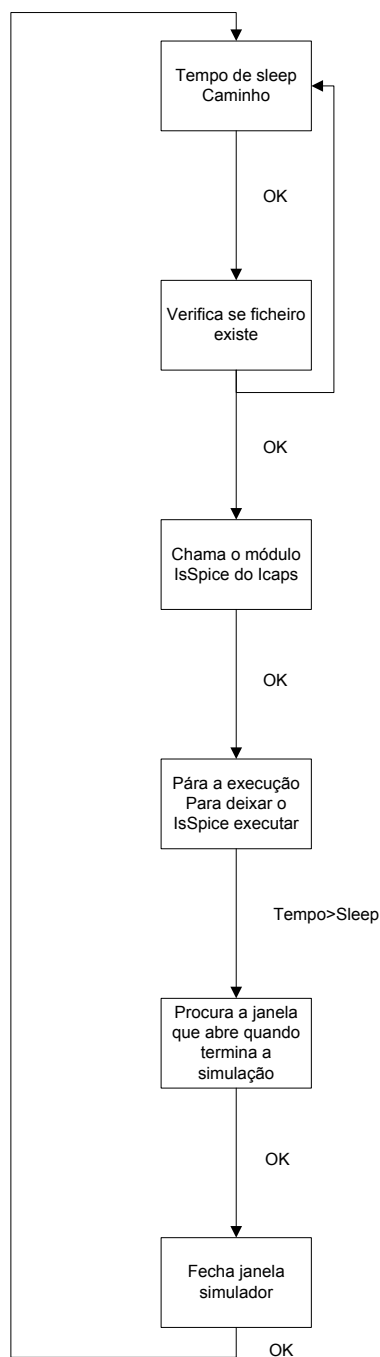
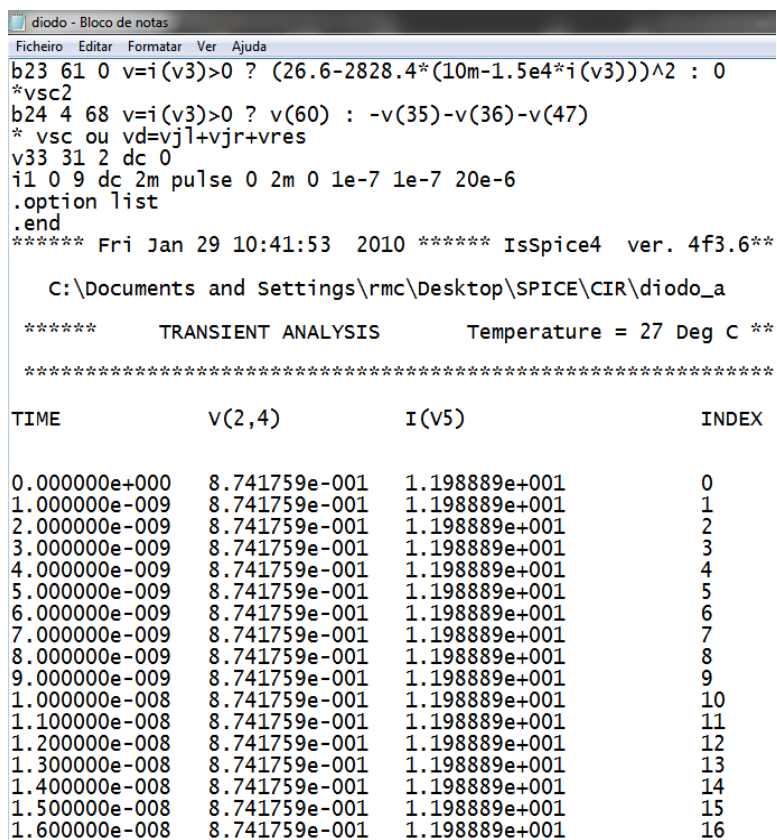


Figura 5-30 - Fluxograma da função “Simulação”.

Com a simulação concluída com sucesso, o programa ICAPS, cria um ficheiro de texto, com a extensão “.out” onde surgem dos dados resultantes da simulação. A Figura 5-31 apresenta um trecho do ficheiro de saída depois de uma simulação.



```

diodo - Bloco de notas
Ficheiro Editar Formatar Ver Ajuda
b23 61 0 v=i(v3)>0 ? (26.6-2828.4*(10m-1.5e4*i(v3)))^2 : 0
*vsc2
b24 4 68 v=i(v3)>0 ? v(60) : -v(35)-v(36)-v(47)
* vsc ou vd=vjl+vjr+vres
v33 31 2 dc 0
i1 0 9 dc 2m pulse 0 2m 0 1e-7 1e-7 20e-6
.option list
.end
***** Fri Jan 29 10:41:53 2010 ***** IsSpice4 ver. 4f3.6**

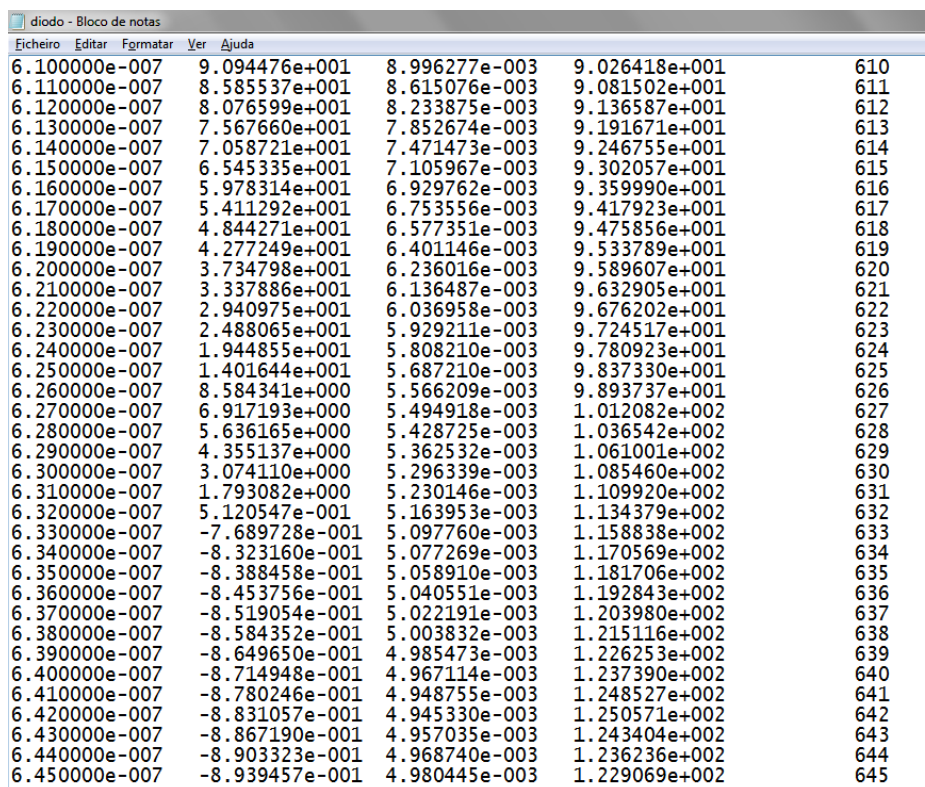
C:\Documents and Settings\rmc\Desktop\SPICE\CIR\diodo_a

*****      TRANSIENT ANALYSIS      Temperature = 27 Deg C **
*****
TIME              V(2,4)              I(V5)              INDEX
0.000000e+000      8.741759e-001      1.198889e+001      0
1.000000e-009      8.741759e-001      1.198889e+001      1
2.000000e-009      8.741759e-001      1.198889e+001      2
3.000000e-009      8.741759e-001      1.198889e+001      3
4.000000e-009      8.741759e-001      1.198889e+001      4
5.000000e-009      8.741759e-001      1.198889e+001      5
6.000000e-009      8.741759e-001      1.198889e+001      6
7.000000e-009      8.741759e-001      1.198889e+001      7
8.000000e-009      8.741759e-001      1.198889e+001      8
9.000000e-009      8.741759e-001      1.198889e+001      9
1.000000e-008      8.741759e-001      1.198889e+001      10
1.100000e-008      8.741759e-001      1.198889e+001      11
1.200000e-008      8.741759e-001      1.198889e+001      12
1.300000e-008      8.741759e-001      1.198889e+001      13
1.400000e-008      8.741759e-001      1.198889e+001      14
1.500000e-008      8.741759e-001      1.198889e+001      15
1.600000e-008      8.741759e-001      1.198889e+001      16

```

Figura 5-31 - Trecho do ficheiro Diodo.out.

Como se pode ver pela análise da Figura 5-31, os resultados são apresentados em colunas, com o nome da variável por cima. Analisando com mais detalhe o ficheiro de saída criado pelo IsSpice, verificou-se que este possuía um “bug” na apresentação de dados. Na Figura 5-32, apresenta-se o “bug” referido. Como é possível verificar existe uma diferença na apresentação dos dados em colunas quando estes assumem valores negativos.



Ficheiro	Editar	Formatar	Ver	Ajuda
6.100000e-007	9.094476e+001	8.996277e-003	9.026418e+001	610
6.110000e-007	8.585537e+001	8.615076e-003	9.081502e+001	611
6.120000e-007	8.076599e+001	8.233875e-003	9.136587e+001	612
6.130000e-007	7.567660e+001	7.852674e-003	9.191671e+001	613
6.140000e-007	7.058721e+001	7.471473e-003	9.246755e+001	614
6.150000e-007	6.545335e+001	7.105967e-003	9.302057e+001	615
6.160000e-007	5.978314e+001	6.929762e-003	9.359990e+001	616
6.170000e-007	5.411292e+001	6.753556e-003	9.417923e+001	617
6.180000e-007	4.844271e+001	6.577351e-003	9.475856e+001	618
6.190000e-007	4.277249e+001	6.401146e-003	9.533789e+001	619
6.200000e-007	3.734798e+001	6.236016e-003	9.589607e+001	620
6.210000e-007	3.337886e+001	6.136487e-003	9.632905e+001	621
6.220000e-007	2.940975e+001	6.036958e-003	9.676202e+001	622
6.230000e-007	2.488065e+001	5.929211e-003	9.724517e+001	623
6.240000e-007	1.944855e+001	5.808210e-003	9.780923e+001	624
6.250000e-007	1.401644e+001	5.687210e-003	9.837330e+001	625
6.260000e-007	8.584341e+000	5.566209e-003	9.893737e+001	626
6.270000e-007	6.917193e+000	5.494918e-003	1.012082e+002	627
6.280000e-007	5.636165e+000	5.428725e-003	1.036542e+002	628
6.290000e-007	4.355137e+000	5.362532e-003	1.061001e+002	629
6.300000e-007	3.074110e+000	5.296339e-003	1.085460e+002	630
6.310000e-007	1.793082e+000	5.230146e-003	1.109920e+002	631
6.320000e-007	5.120547e-001	5.163953e-003	1.134379e+002	632
6.330000e-007	-7.689728e-001	5.097760e-003	1.158838e+002	633
6.340000e-007	-8.323160e-001	5.077269e-003	1.170569e+002	634
6.350000e-007	-8.388458e-001	5.058910e-003	1.181706e+002	635
6.360000e-007	-8.453756e-001	5.040551e-003	1.192843e+002	636
6.370000e-007	-8.519054e-001	5.022191e-003	1.203980e+002	637
6.380000e-007	-8.584352e-001	5.003832e-003	1.215116e+002	638
6.390000e-007	-8.649650e-001	4.985473e-003	1.226253e+002	639
6.400000e-007	-8.714948e-001	4.967114e-003	1.237390e+002	640
6.410000e-007	-8.780246e-001	4.948755e-003	1.248527e+002	641
6.420000e-007	-8.831057e-001	4.945330e-003	1.250571e+002	642
6.430000e-007	-8.867190e-001	4.957035e-003	1.243404e+002	643
6.440000e-007	-8.903323e-001	4.968740e-003	1.236236e+002	644
6.450000e-007	-8.939457e-001	4.980445e-003	1.229069e+002	645

Figura 5-32 - Trecho2 do ficheiro diodo.out.

Desta forma implementou-se uma função, que aproveitando a disposição dos dados em colunas, os extraísse e guardasse em ficheiros de texto, um para cada variável. Para tal foi necessário corrigir o erro na apresentação de valores, tendo-se verificado de extrema importância a detecção deste problema.

Esta função faz ainda duas importantes alterações. Por um lado substituiu os pontos “.” por vírgulas “,” e por outro lado substituiu e calcula o expoente “e” para se poder utilizar nas componentes gráficas que o software de desenvolvimento possui.

O fluxograma da função implementada é apresentado na Figura 5-33.

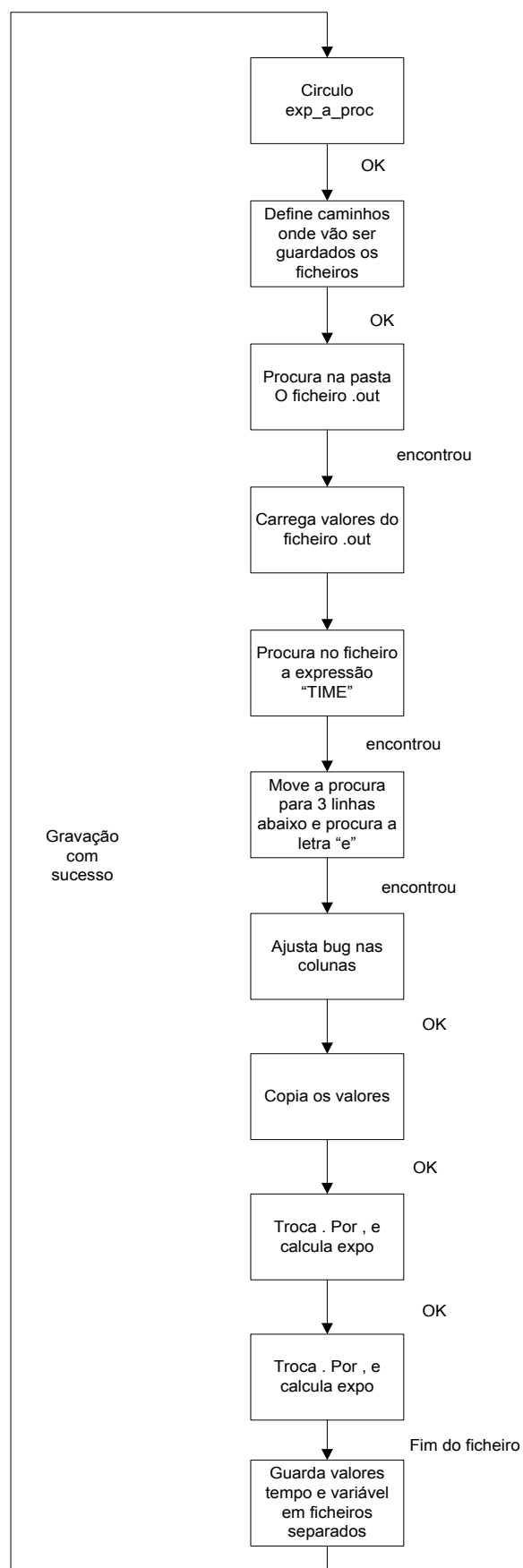


Figura 5-33 - Fluxograma da função "Carrega_valores_saída".

5.7 - Representação gráfica

O Delphi 7 possui componentes para representações gráficas bidimensionais no seu pacote de instalação primário. No entanto, no que diz respeito aos gráficos tridimensionais, é necessário instalar componentes extras.

Assim, fez-se uso de um componente extra, intitulado de SDL Suite 7.0 que depois de instalado adiciona uma série de funcionalidades ao programa de desenvolvimento. No que diz respeito aos gráficos bidimensionais, utilizaram-se os “Rchart” que possuem as seguintes características:

- Permitem desenhar várias formas de onda no mesmo gráfico, sendo possível controlar a cor e a espessura destas;
- Possibilidade de personalização da área do gráfico;
- Possibilidade de copiar a imagem visualizada;
- Elevado grau de eventos associados ao rato;
- Inserção de linhas verticais (marcadores);
- Vários tipos de Zoom e movimentação dos gráficos;

Em relação aos gráficos tridimensionais, fez-se uso de “Rot3D” que possui as seguintes particularidades:

- Capacidade de rotação dos gráficos;
- Capacidade de colocar etiquetas nos eixos dos gráficos;
- Possibilidade de desenhar superfícies tridimensionais, através da união de pontos por rectas;

5.8 - Instalação de um único ficheiro

Na fase final do desenvolvimento da interface surgiu a necessidade de testar a interface em vários sistemas operativos da MS, com resoluções de ecrãs diferentes. Desta forma e sempre que se pretendia fazer esses testes, era necessário copiar a pasta de simulação do ICAPS, o executável da interface e a pasta com os ficheiros de simulação. Assim, e fazendo uso de uma ferramenta - InnoSetup - foi possível transformar essas três pastas num ficheiro único de instalação. Para se criar um ficheiro executável basta correr um assistente de configuração que permite parametrizar os atalhos a disponibilizar, a pasta de instalação, as diferentes linguagens disponíveis, entre outros parâmetros, criando assim um ficheiro único de instalação. Este ficheiro é posteriormente utilizado para se instalar o programa em diferentes sistemas operativos.

5.9 - Gravação para PDF

Este ponto é importante uma vez que permite guardar os resultados obtidos pelas simulações e as respectivas explicações teóricas. Esta vantagem pode ser também utilizada pelo docente, uma vez que possibilita que durante uma aula de explicação sobre o funcionamento dos dispositivos semicondutores, possa efectuar várias simulações, adicionar as respectivas explicações sobre o funcionamento e posteriormente disponibilizar os ficheiros obtidos para posterior consulta por parte dos estudantes.

Para efectuar a gravação num formato “.pdf” foi necessário recorrer a um componente extra do Borland Delphi 7, quickpdf.

5.10 - Conclusões

Após o término deste capítulo deve-se realçar a importância das regras apresentadas para se poder obter uma interface intuitiva, clara, sucinta e que cumpra os requisitos pretendidos. A definição de objectos que partilhem os mesmos atributos e métodos, as classes, permitem a reutilização de código, facilitando não só o trabalho do programador actual mas também de outro que seja incumbido de dar continuidade ao trabalho. Assim, implementaram-se e apresentaram-se as classes da interface desenvolvida e os diagramas UML que permitem modelizar a aplicação. Na parte final do capítulo apresentaram-se algumas funções de tratamento de dados e de simulação que possibilitaram atingir os objectivos propostos. O uso da ferramenta InnoSetup, permite transformar um conjunto de pastas num único ficheiro de instalação, dando-lhe um ar mais profissional e organizado. No que diz respeito ao componente extra utilizado para gravação no formato “.pdf”, revelou-se bastante interessante permitindo o posterior estudo por parte dos estudantes.

Capítulo 6

Demonstração e Validação

6.1 - Introdução

Neste capítulo é apresentada a interface desenvolvida. No ponto 6.2 faz-se uma demonstração do funcionamento e das potencialidades da interface desenvolvida e da forma de interacção desta com o utilizador. Esta demonstração é apenas realizada para o circuito do díodo uma vez que os processos para o circuito do IGBT são os mesmos. Por fim, no ponto 6.3, utiliza-se um exemplo clássico do funcionamento do díodo com o intuito de validar a utilização da interface.

6.2 - Demonstração

Na Figura 6-1 apresenta-se o painel de arranque do SWEESC (Software Educacional para o Ensino de Semicondutores). A evolução da barra de progresso significa que estão a ser verificados os caminhos pré-definidos para o simulador e para os ficheiros de simulação.

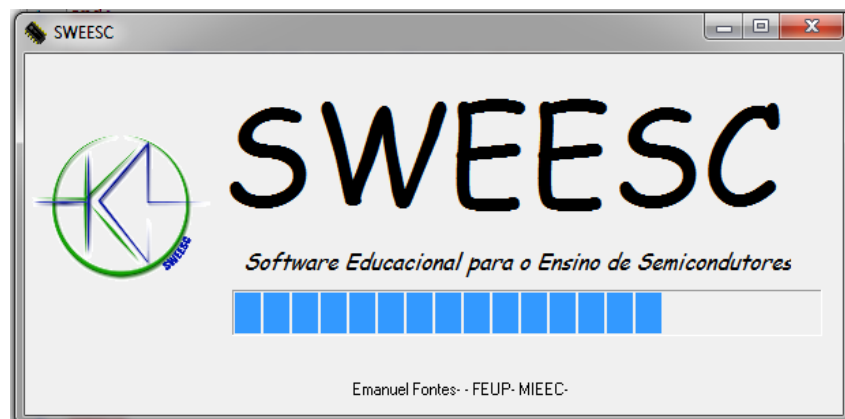


Figura 6-1 - SWEESC - Janela de arranque.

A Figura 6-2 apresenta a janela de apresentação da interface. Esta janela surge quando a verificação dos caminhos é concluída, ou seja, quando a janela de arranque se fecha. Aqui, o utilizador toma consciência das potencialidades gerais da interface. O menu, colocado no topo das janelas, é característico de quase todas as aplicações usadas diariamente, desde editores de texto, a programas complexos, que na sua maioria possuem um menu como o que foi desenvolvido.

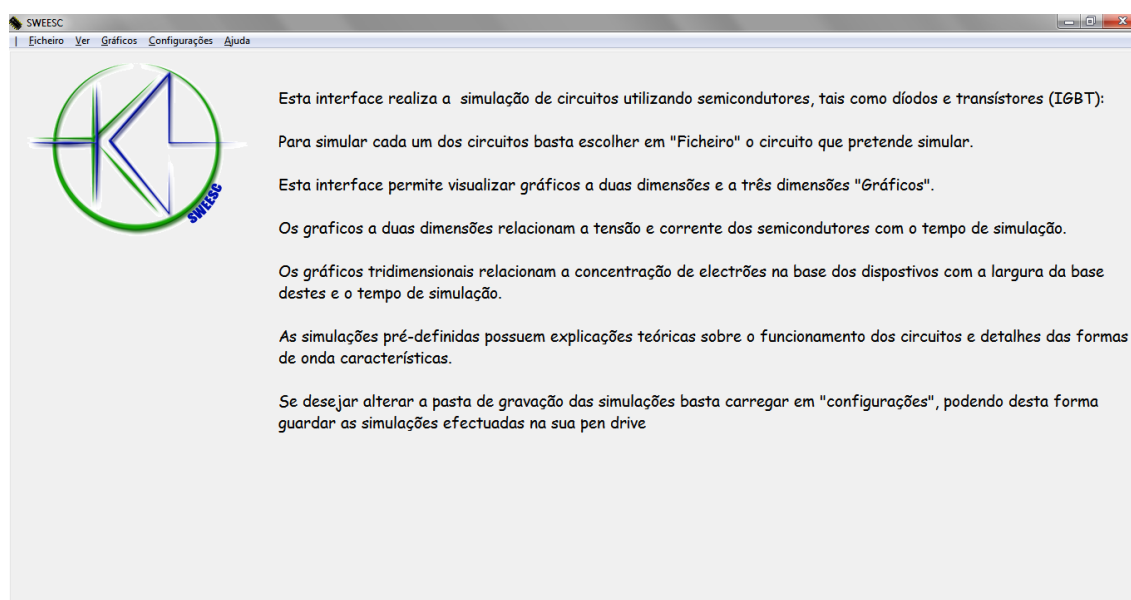


Figura 6-2 - SWEESC - Janela de apresentação.

Para aceder à janela do circuito do díodo, basta fazer a seguinte combinação: “Ficheiro”, “Abrir”, “Circuitos”, “Díodo”. A Figura 6-3 apresenta a janela com o circuito do díodo. Como se pode verificar pela análise da imagem, a barra de menus continua no topo da página, permitindo uma navegação constante entre as diferentes janelas que compõem a interface. Existe ainda uma barra de atalhos rápidos permitindo ao utilizador, por exemplo, simular o circuito (Botão Simular) ou escolher gráficos de simulações pré-definidas (Botão Escolher Gráfico). Os botões “Gráficos 2D” e “Gráficos 3D” encontram-se inactivos, uma vez que não foi realizada nenhuma simulação desde que o utilizador arrancou a interface. Esta decisão tem real interesse quando se apresentar a janela de escolha dos gráficos de simulações pré-definidas. Deve-se ainda salientar que estes botões de atalhos tem uma imagem associada que permite ao utilizador interpretar rapidamente a função de cada um.

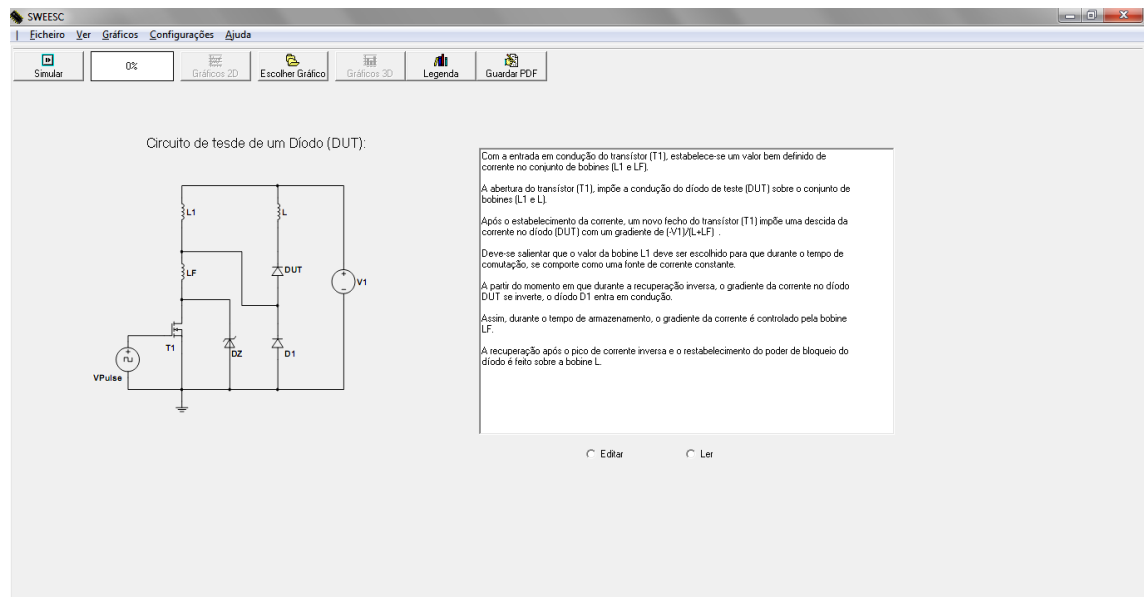


Figura 6-3 - SWEESC - Circuito do diodo.

Como se pode observar na Figura 6-3, o circuito do diodo é acompanhado de uma explicação teórica sobre o funcionamento do mesmo. Esta explicação pode ser editada, no caso de o estudante pretender adicionar uma explicação adicional. Existe ainda um botão que permite guardar a imagem do circuito e a explicação teórica para o formato .pdf e um botão para ver a legenda do circuito, ou seja, o componente, a sua grandeza e um valor típico.

A Figura 6-4 apresenta a janela de legenda do circuito. No que diz respeito ao pdf gerado, este é apresentado no Anexo A desta dissertação.

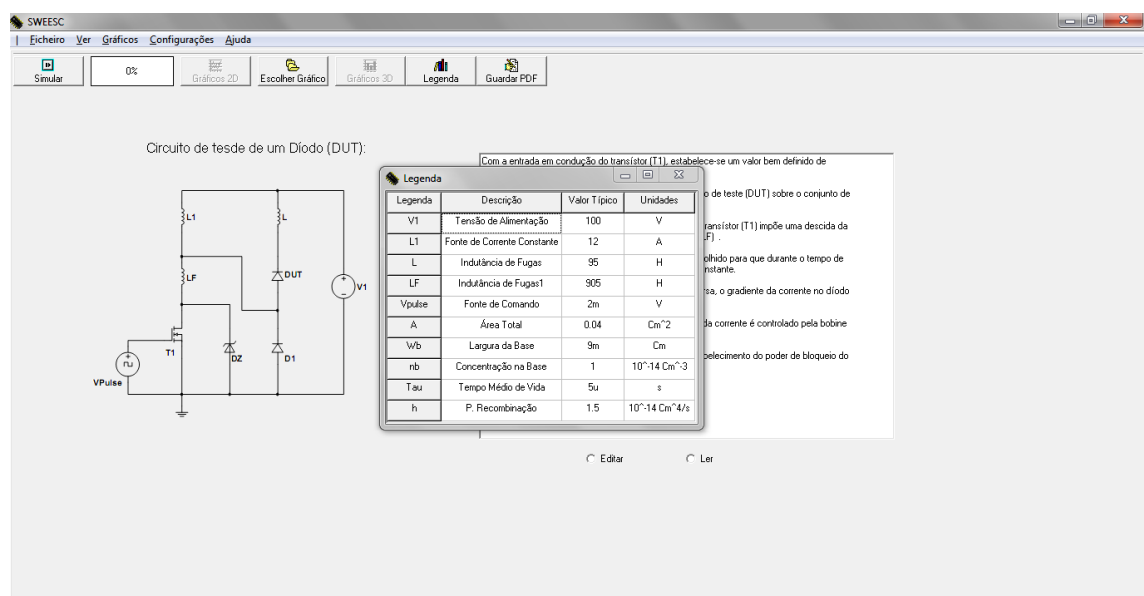


Figura 6-4 - SWEESC - Legenda do circuito.

Nas Figura 6-5 e Figura 6-6 apresenta-se uma particularidade da interface desenvolvida. Quando se pretender verificar o valor pré-definido dos componentes do circuito, basta passar o cursor do rato por cima destes e surgirá, ao lado do componente, o valor pré-definido.

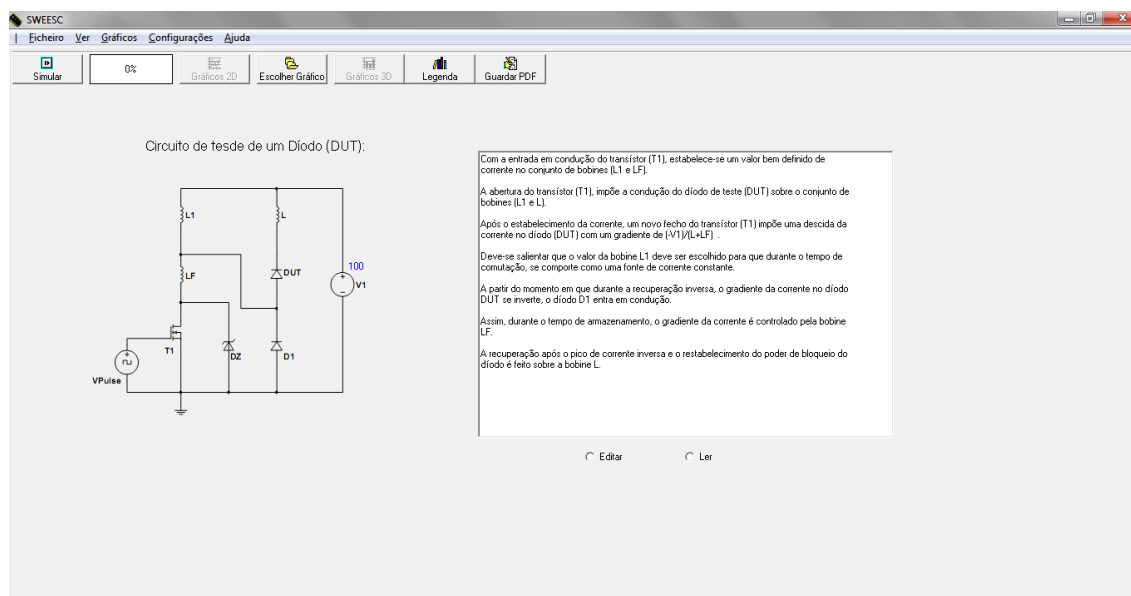


Figura 6-5 - SWEESC - Circuito do díodo - Legenda dos componentes - V1.

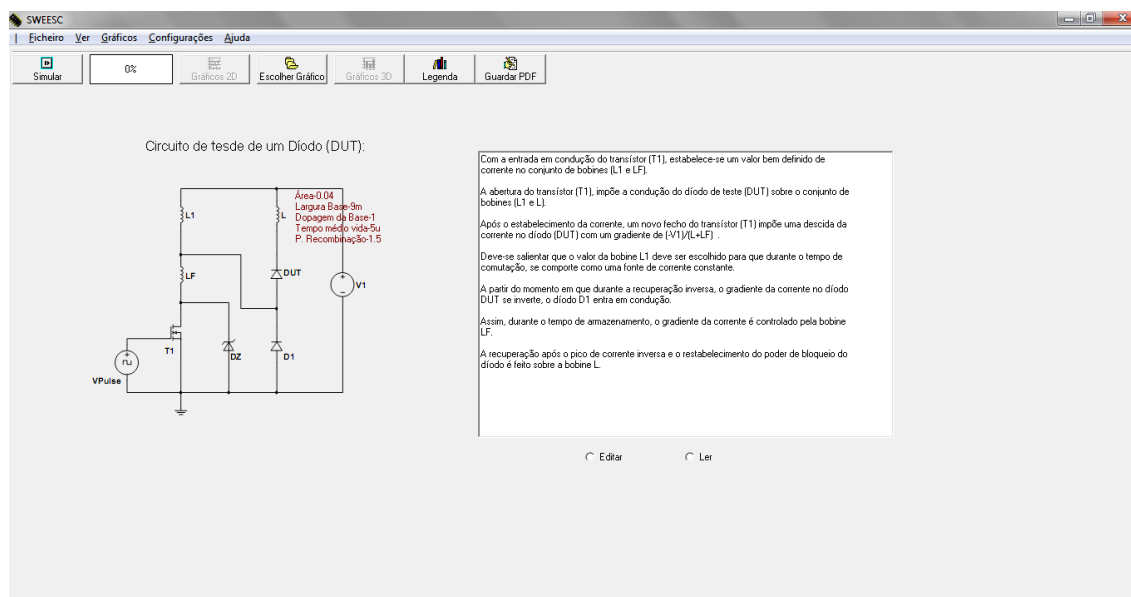


Figura 6-6 - SWEESC - Circuito do díodo - Legenda dos componentes - DUT.

Se o utilizador pretender alterar este valor pré-definido, basta clicar no botão esquerdo do rato, em cima do componente e surgirá uma janela, onde é possível não só escolher outro

valor para o componente, como também alterar os valores pré-definidos. Deve-se notar que a barra de menu continua visível permitindo ao utilizador uma navegação rápida e eficaz.

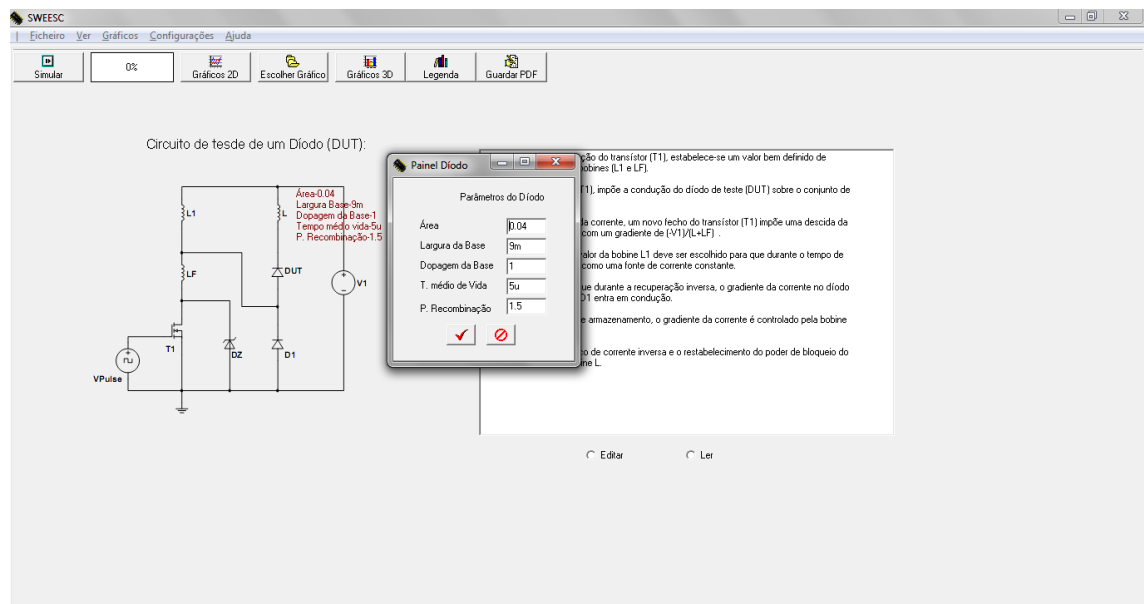


Figura 6-7 - SWEESC - Circuito do díodo - Painel de alteração de valores.

No caso de o utilizador pretender simular o circuito, basta, para tal, carregar no botão “Simular”. Ao carregar no botão “Simular”, é enviada uma ordem para executar o módulo IsSpice do programa ICAPS. Na Figura 6-8 é possível identificar as diferentes janelas que surgem na simulação de um circuito e que foram apresentadas no Capítulo 4, secção 4.2. Estas janelas podem ser ocultadas, tornando o acto de simulação invisível para o utilizador.

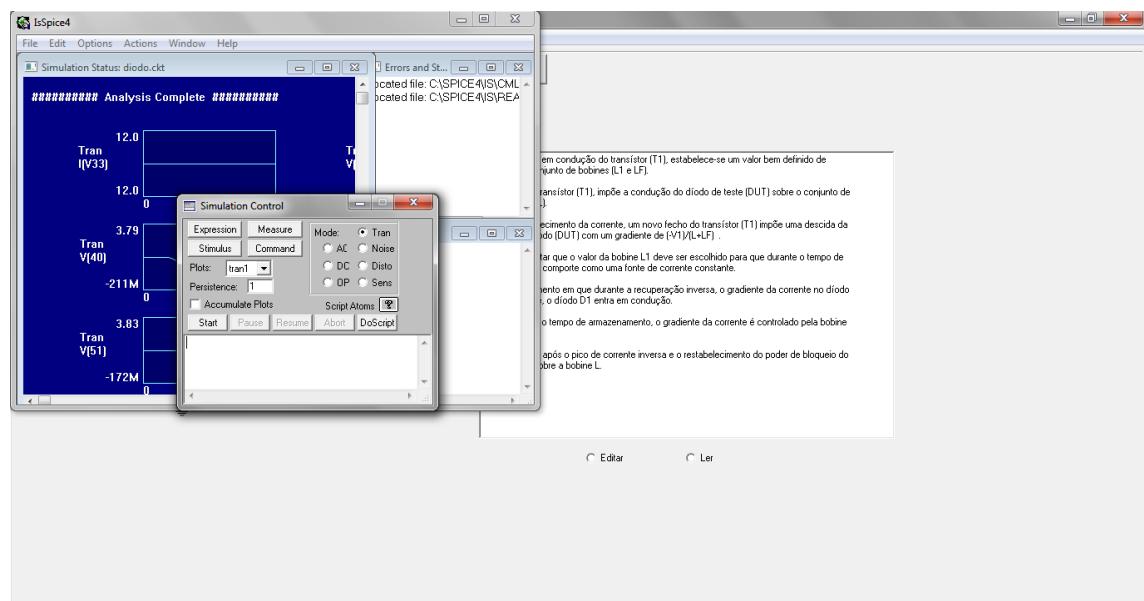


Figura 6-8 - SWEESC - Simulação de um circuito.

Após a simulação do circuito, é realizada a extracção dos dados de saída resultantes da simulação. De forma a mostrar ao utilizador a evolução desse processo, existe um pequeno painel com a evolução em percentagem. Quando o processo de simulação e extracção dos dados de saída terminar, surgirá uma janela de confirmação de que o processo foi concluído com sucesso - Figura 6-9 .

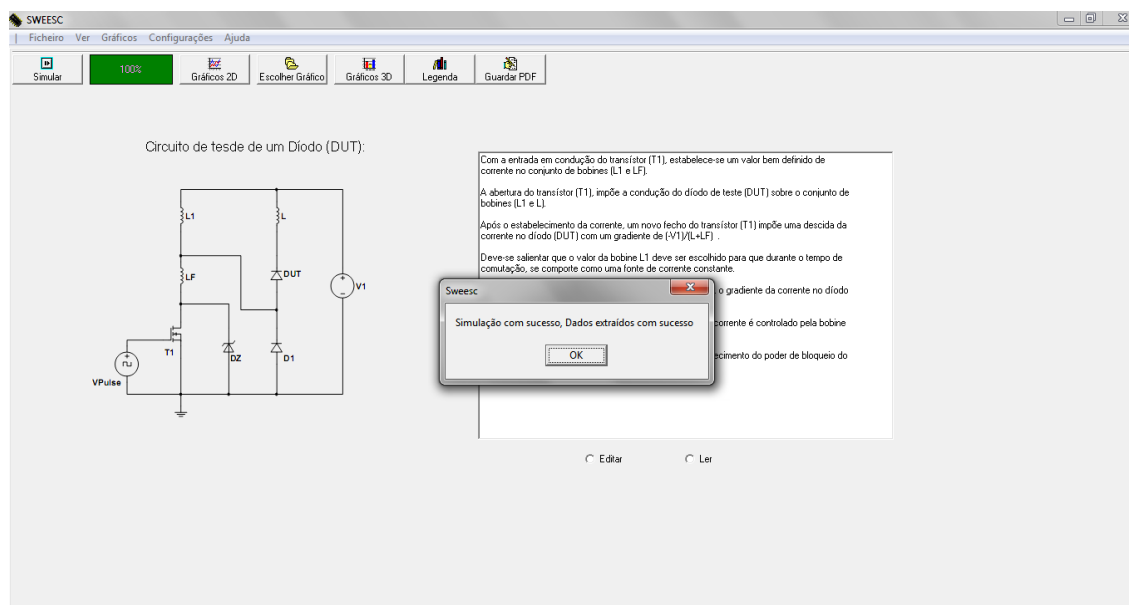


Figura 6-9 - SWEESC - Simulação do diodo - Janela de confirmação de simulação bem sucedida.

Depois de confirmar a mensagem informativa, a interface permite que o utilizador veja os gráficos correspondentes à simulação efectuada, activando os botões "Gráficos 2D" e "Gráficos 3D".

Assim, ao carregar no botão "Gráficos 2D" surgirá uma janela - Figura 6-10 - com a primeira linha da tabela de simulações preenchida. Esta opção permite que o utilizador possa adicionar simulações pré-definidas, comparando os parâmetros que foram alterados e desta forma poder visualizar as formas de ondas características de cada simulação.

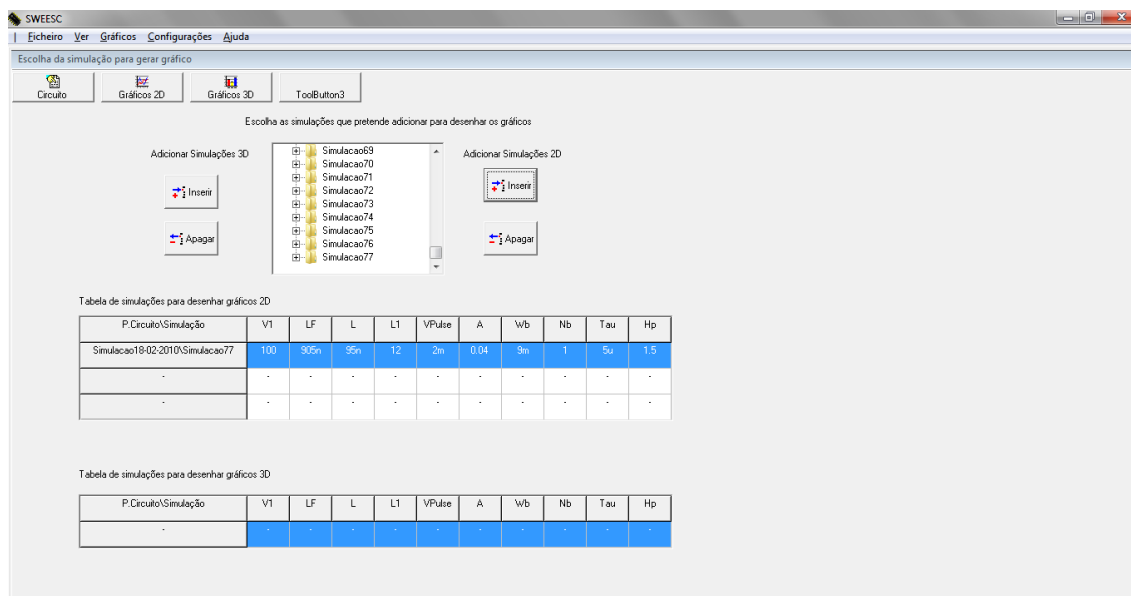


Figura 6-10 - SWEESC- Janela de escolha dos gráficos a simular.

Esta janela é extremamente útil porque permite adicionar um máximo de três simulações para as representações 2D e apenas uma simulação para as representações 3D, permitindo a comparação de novas simulações com outras previamente guardadas. Para gerar os gráficos a duas dimensões basta carregar no botão “Gráficos 2D”.

Na Figura 6-11 apresenta-se a janela de visualização dos gráficos 2D. Esta janela é composta pelos gráficos de tensão e corrente em função do tempo, e por uma tabela onde se apresentam valores máximos e mínimos, e os tempos em que estes ocorrem e por fim os parâmetros do diodo utilizado na simulação.

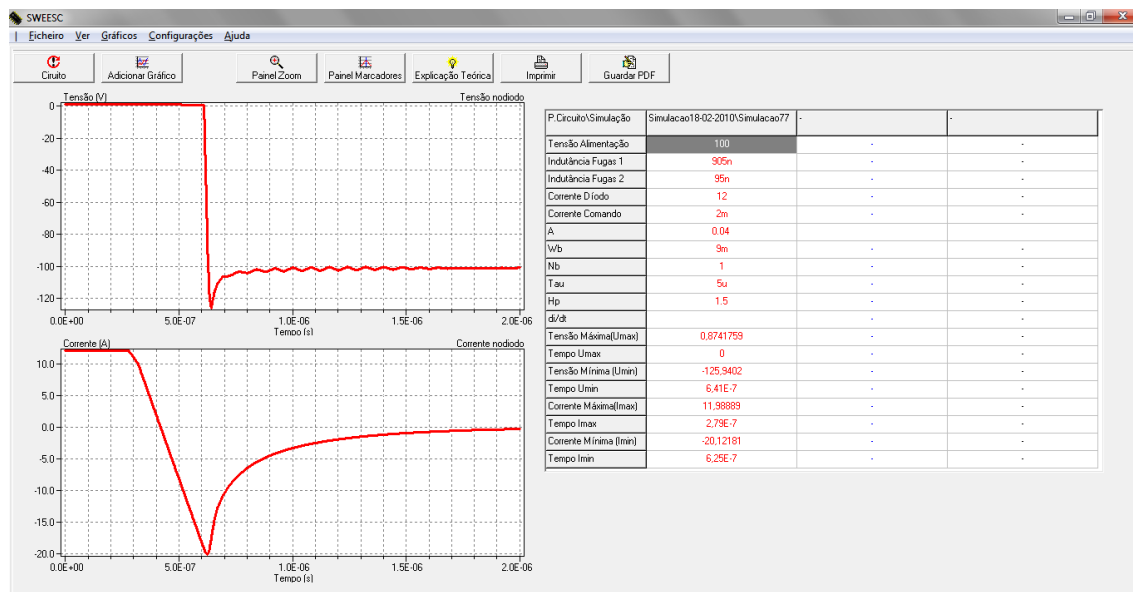


Figura 6-11 - SWEESC - Janela de visualização dos gráficos 2D.

Na Figura 6-12 apresentam-se os resultados de duas simulações onde se variou um parâmetro do diodo (tempo médio de vida).

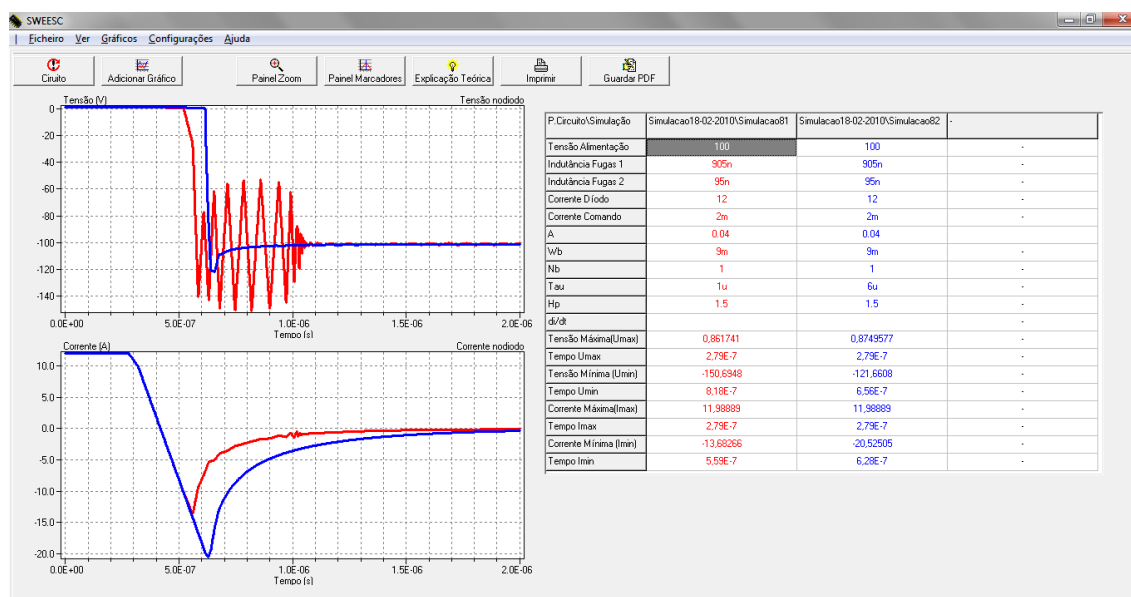


Figura 6-12 - SWEESC- Janela de visualização dos gráficos 2D (2 simulações).

Ao nível dos gráficos 2D foram implementadas várias funcionalidades que permitem melhorar a visualização e a interação entre o utilizador e a interface. Desta forma, implementou-se uma janela de zoom, que permite, não só aplicar ampliar ou diminuir a

janela de visualização, mas também movimentar os gráficos quer horizontalmente quer verticalmente. Para adicionar ainda mais potencialidades à análise gráfica implementou-se um painel que permite colocar barras verticais e horizontais nos gráficos de tensão e corrente, possibilitando, desta forma, a medição de tempos ou grandezas. Esses painéis são apresentados na Figura 6-13 e na Figura 6-14 respectivamente.

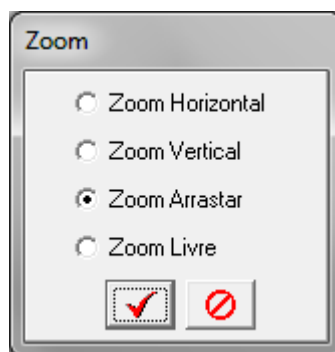


Figura 6-13 - SWEESC - Painel zoom.

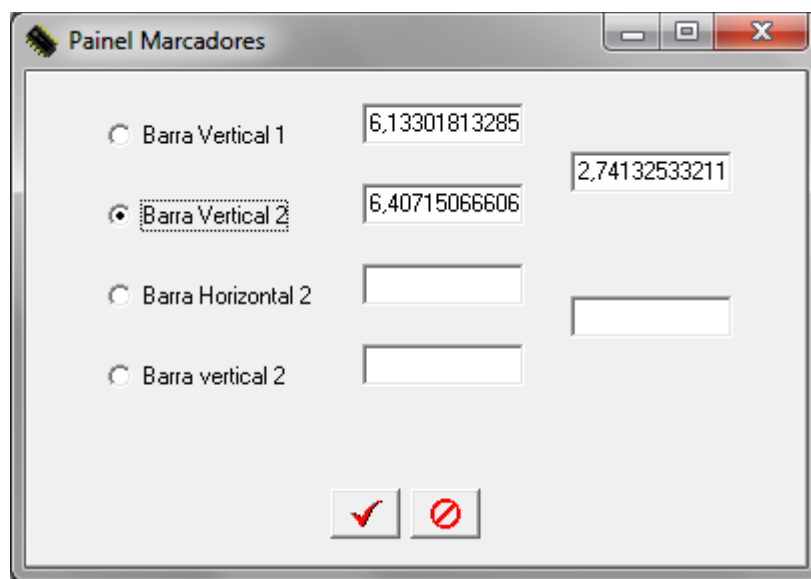


Figura 6-14 - SWEESC - Painel Marcadores.

Utilizando o painel de zoom consegue-se ampliar uma determinada zona de um gráfico e com o recurso ao painel dos marcadores consegue-se de uma forma eficiente medir tempos ou grandezas.

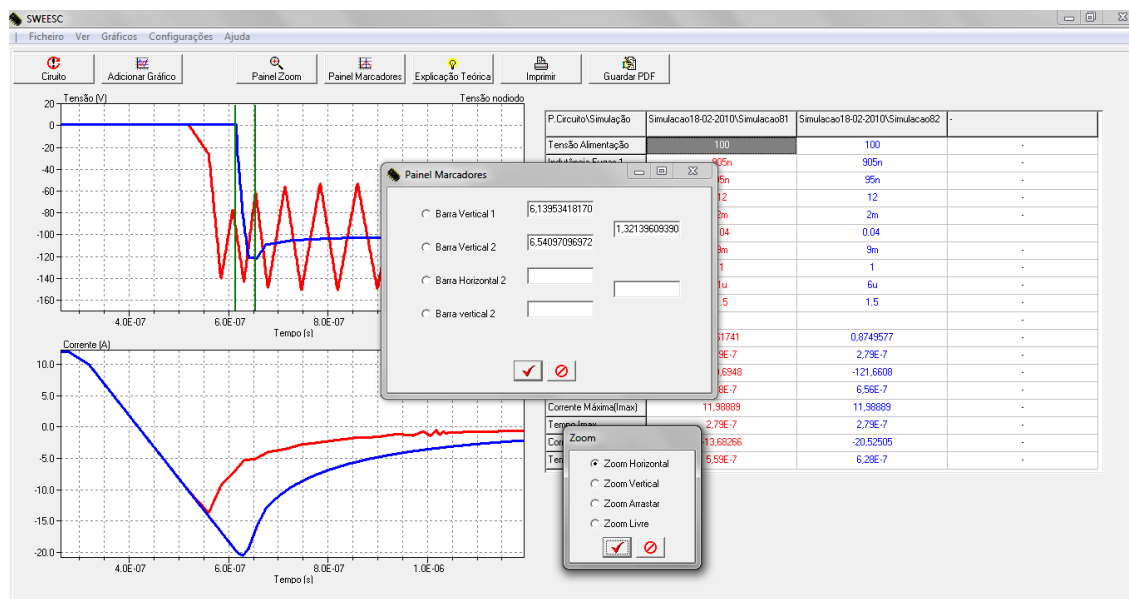


Figura 6-15 - SWEESC - Gráficos 2D com utilização dos painéis auxiliares.

Neste ponto deve-se salientar que é possível guardar os gráficos obtidos e a respectiva explicação teórica para um ficheiro com o formato “.pdf” . Este ficheiro é apresentado no Anexo A desta dissertação.

Ao nível da implementação dos gráficos tridimensionais e tal como nos gráficos a bidimensionais, existe uma janela onde é possível escolher a simulação pré-definida que se pretende visualizar - Figura 6-16.

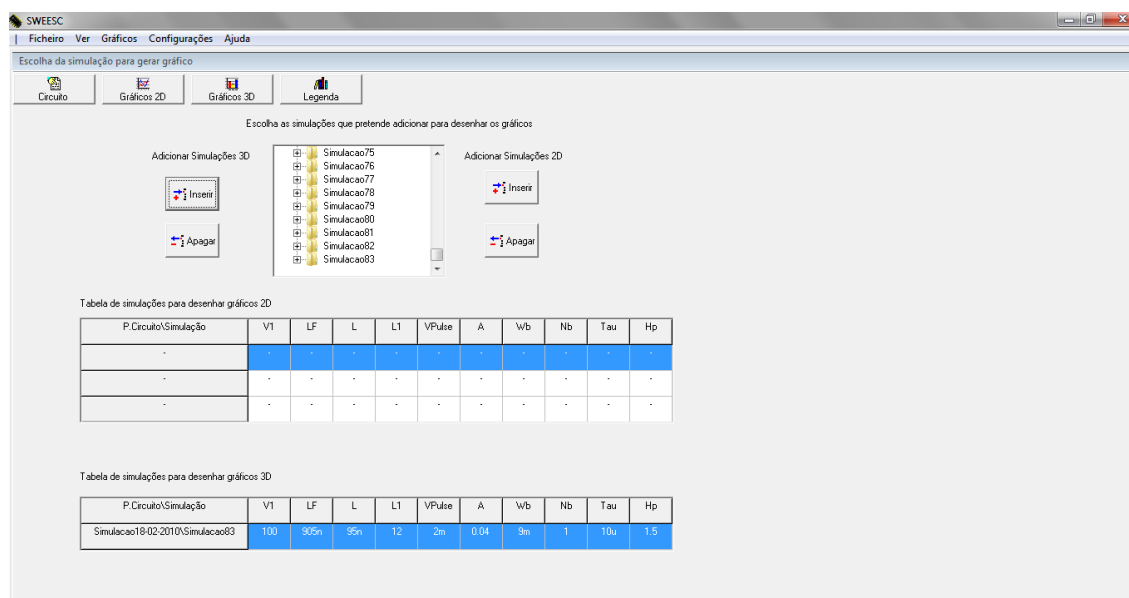


Figura 6-16 - SWEESC - Escolha da simulação para gerar Gráfico 3D.

Esta janela aparece da mesma forma se depois da simulação se carregar no botão “gráficos 3D”, sendo a tabela preenchida com os valores da simulação actual.

A Figura 6-17 representa a visualização a três dimensões. Os gráficos tridimensionais permitem visualizar a concentração de electrões em função do tempo de simulação e da largura da base do semiconductor. Neste ponto optou-se apresentar uma superfície a rodar automaticamente (1) e duas superfícies representativas da variação da concentração com a largura da base (2) e da variação da largura da base com o tempo (3).

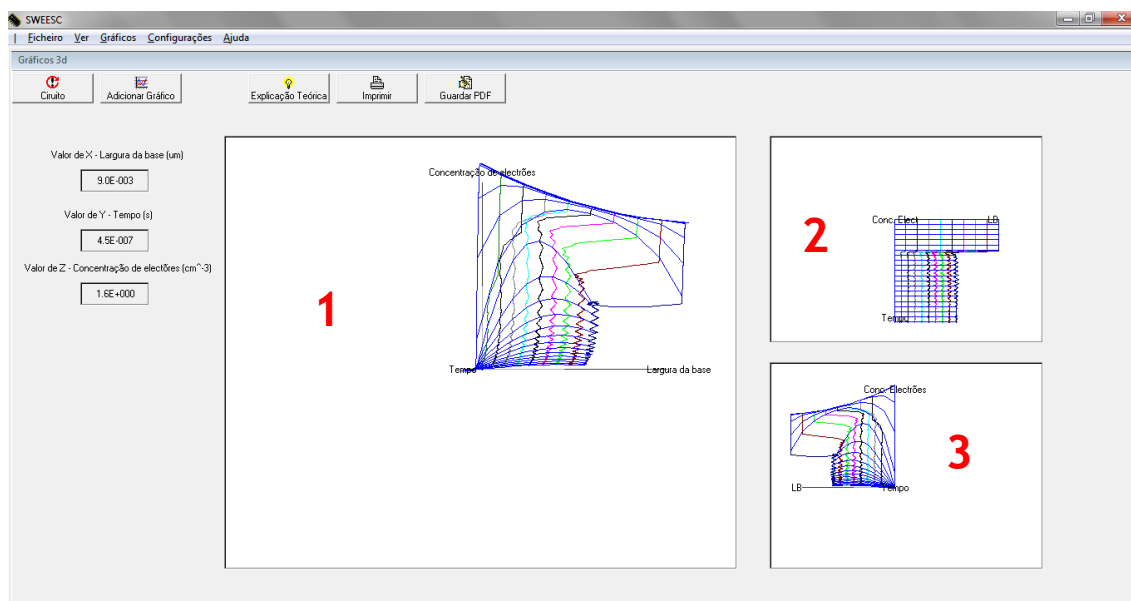


Figura 6-17 - SWEESC - Visualização tridimensional.

Tal como na visualização a duas dimensões é possível editar ou adicionar a explicação teórica e guardar num ficheiro com a extensão “.pdf”, ou imprimir directamente para uma impressora previamente instalada no computador.

No canto superior esquerdo aparecem os valores conforme a posição do ponteiro do rato. Optou-se por esta solução, uma vez que os gráficos têm a limitação de não ser possível definir legenda para pontos situados nos eixos. Relativamente à manipulação do gráfico (1), é possível rodá-lo em qualquer direcção, mantendo-se os gráficos(2 e 3) estáticos.

Para terminar a demonstração da aplicação apresenta-se a característica da interface que permite ao utilizador alterar os caminhos pré-definidos. Para tal basta que o utilizador carregue no botão “Configurações - Caminhos”.

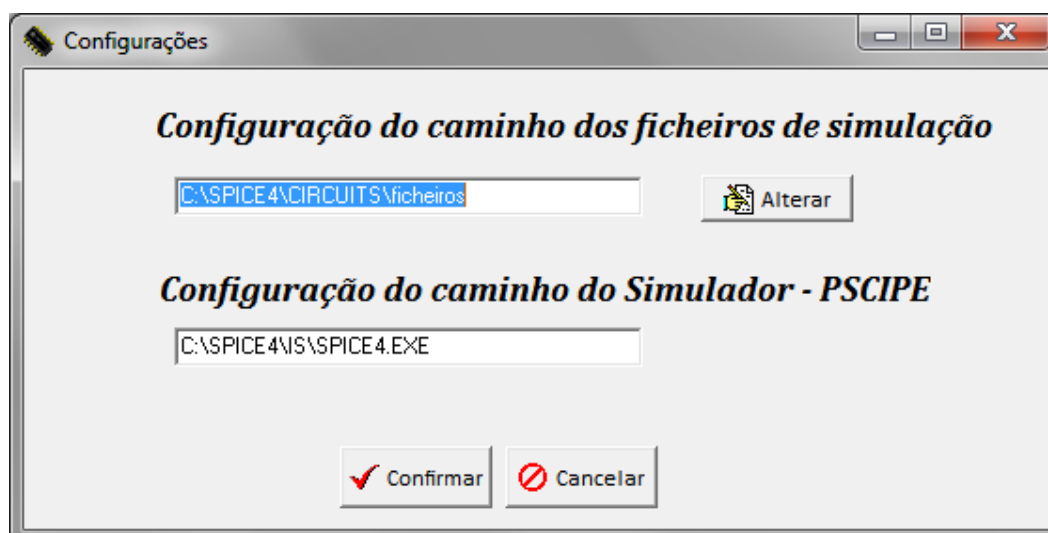


Figura 6-18 - SWEESC - Janela de visualização dos caminhos do sistema.

Para alterar os caminhos pré-definidos basta carregar no botão “Alterar” e surgirá uma janela - Figura 6-19 onde mostra a vista em árvore dos ficheiros do sistema. Desta forma é permitido ao utilizador gravar as simulações no seu disco pessoal.

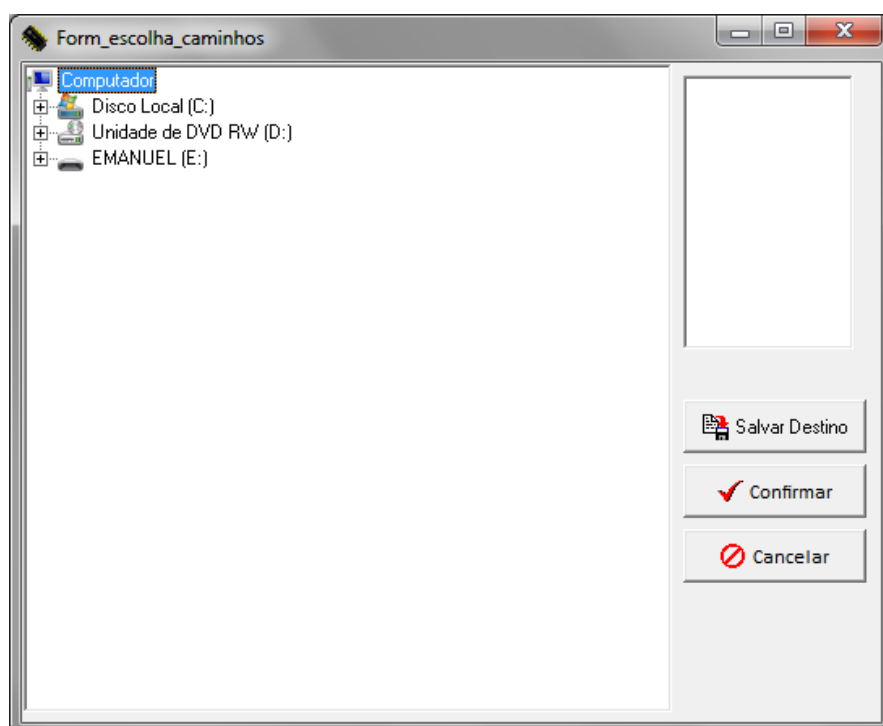


Figura 6-19 - SWEESC - Janela de escolha do caminho.

6.3 - Validação

Neste ponto procede-se à validação da interface desenvolvida. Para tal recorreu-se a um exemplo clássico da teoria dos semicondutores.

Da teoria dos díodos é de esperar que o aumento do tempo médio de vida provoque um aumento do tempo de recuperação inversa. Este fenómeno acontece porque, ao aumentar o tempo médio de vida de um diodo, há um acréscimo de carga disponível na zona N⁺, que por sua vez incrementa o tempo de recombinação entre electrões e lacunas provocando assim um aumento do tempo de recuperação inversa.

A Figura 6-20 apresenta três simulações onde se variou o tempo médio de vida do diodo de acordo com a Tabela 4-1.

Tabela 6-1 Tabela da variação do tempo médio de vida.

Tempo médio de vida	Gráfico
1E-6 s	Vermelho
6E-6 s	Azul
10E-6 s	Preto

Como é possível verificar, no gráfico da corrente (o inferior), o tempo de recuperação inversa é maior no gráfico a preto (maior tempo médio de vida) e menor no gráfico a azul (menor tempo médio de vida).

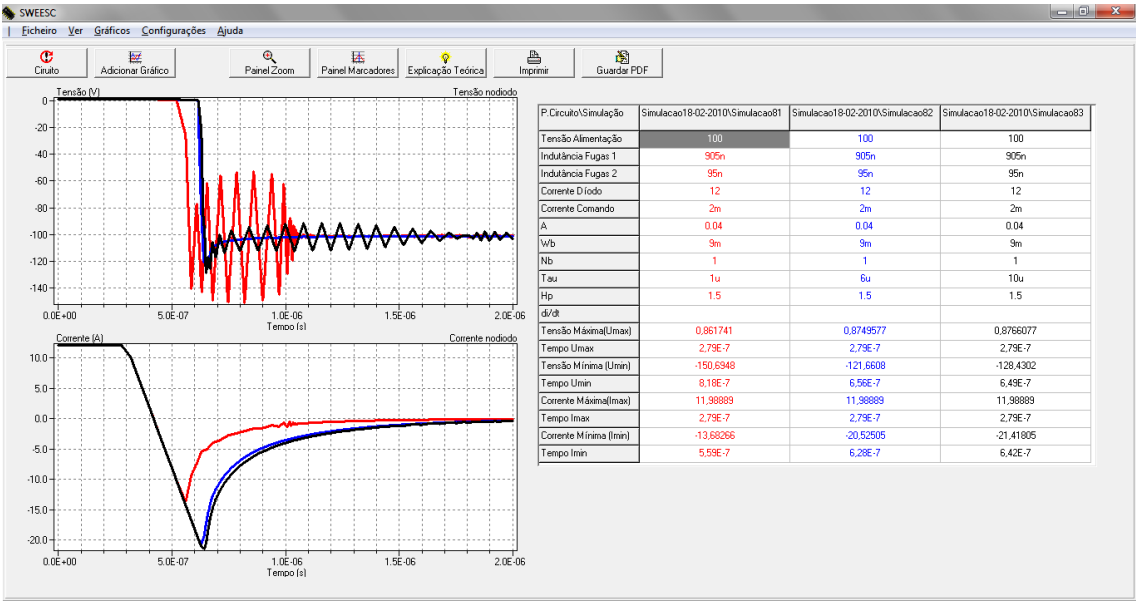


Figura 6-20 - Validação da interface - Variação do tempo médio de vida de um diodo.

Neste ponto, deve-se referir umas das inovações desta interface. Depois de efectuar uma simulação com diferentes parâmetros, é possível escrever ou carregar uma explicação teórica para os efeitos observados. A Figura 6-21 exemplifica essa particularidade.

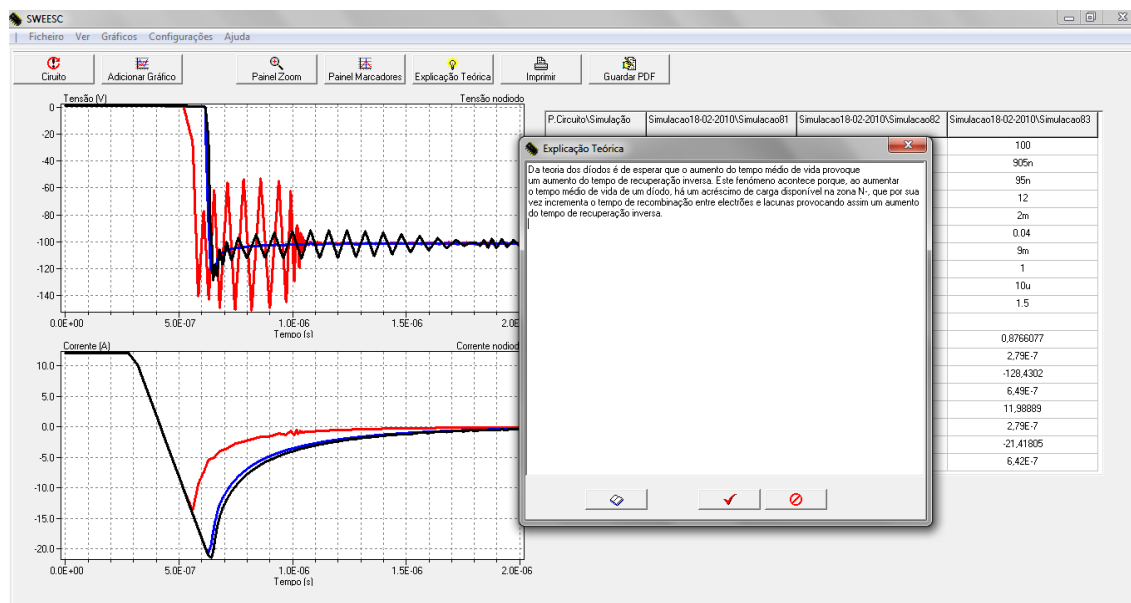


Figura 6-21 - SWEESC - Explicação teórica sobre a variação do tempo médio de vida.

6.4 - Conclusões

Com o término deste capítulo pode-se concluir o correcto funcionamento da interface desenvolvida, como é prova, o exemplo da variação do tempo médio de vida de um diodo. Além disso, pode-se concluir que a simulação recorrendo a modelos híbridos dos semicondutores permite explicar fenómenos associados às propriedades físicas dos semicondutores. Por fim, a interface desenvolvida cumpre todos os requisitos previstos, com a excepção dos gráficos tridimensionais, que muito embora estejam próximos do objectivo que se pretende, este ainda não foi alcançado.

Capítulo 7

Conclusões e Trabalho Futuro

7.1 - Conclusões

Com esta dissertação foi possível perceber o estado de evolução do desenvolvimento dos dispositivos semicondutores, no que diz respeito à progressiva diminuição de tamanho e aumento de velocidade.

Do ponto de vista tecnológico percebeu-se a diferença entre os modelos híbridos e os modelos empíricos dos semicondutores e a importância que os primeiros assumem no estudo das características físicas destes.

No que diz respeito à programação da interface desenvolvida, a utilização de classes foi essencial para estruturar e evitar repetição de código. A implementação com classes permite agrupar objectos com os mesmos atributos e métodos possibilitando desta forma a reutilização de código. A utilização de ficheiros para *backup* de dados é uma solução interessante, permitindo criar um histórico de simulações. As explicações teóricas adicionadas aos gráficos e aos circuitos permitem ao utilizador interpretar os fenómenos ocorridos em simulações pré-definidas, assimilando os conhecimentos sobre a influência dos diversos parâmetros físicos dos semicondutores. As diversas funcionalidades implementadas, “zoom”, “marcadores” nos gráficos, etc, possibilitam uma melhor visualização dos fenómenos ocorridos.

Ao nível do aspecto gráfico, as regras definidas e implementadas revelaram ser uma grande ajuda para realizar uma interface clara, concisa e coerente com os objectivos propostos.

Os objectivos propostos foram alcançados em pleno e ainda foram adicionadas funcionalidades extras, tais como a gravação no formato “.pdf” e a criação de um ficheiro único de instalação.

Por fim, pode-se concluir que uma interface de simulação desenvolvida pode ser uma ferramenta muito útil no apoio ao ensino dos semicondutores em meios académicos.

7.2 - Trabalho Futuro

Em relação a futuros desenvolvimentos pensa-se que seria interessante:

- Avaliar a viabilidade da utilização de uma base de dados em detrimento dos ficheiros de texto.
- Implementar um módulo de introdução de circuitos que criasse automaticamente o ficheiro de simulação;
- Avaliar a viabilidade de exportar os dados no formato “xlsx” (Excel) de forma a confirmar os resultados obtidos com esta interface.
- Avaliar a viabilidade de criar circuitos genéricos e realizar a sua simulação.

Referências

- [1] Y. T. Yeow and C. H. Ling, "Teaching semiconductor device physics with two-dimensional numerical solver", IEEE Transactions on Education, vol. 42, no. 1, pp. 50-58, 1999.
- [2] J. Záček, "Simulation of power semiconductor systems as an educational tool", Microelectronics Journal, vol. 27, no. 2-3, pp. 149-159, 1996.
- [3] R. Santos, R. Chibante, A. Araújo and A. Carvalho, "TESE- A tool for Teaching Semiconductor Theory", Proceedings of 2nd International Conference on Power Engineering, Energy and Electrical Drives (POWERENG'09), pp. 640-644, 18-20 Mar, Lisboa, Portugal, 2009
- [4] IsSPICE4- Simulador utilizado pelo programa de simulação ICAPS. Mais informações: <http://www.intusoft.com/icap.htm#isspice4> [Acedido pela última vez em 25-01-2010].
- [5] ICAPS - Programa de simulação de circuitos analógicos e digitais. Mais informações: <http://www.intusoft.com/products/ICAP4Windows.htm> [Acedido pela última vez em 25-01-2010].
- [6] A. Araújo, "Formulação de um novo método para simulação em circuitos electrónicos de potência", Dissertação de doutoramento em Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto.
- [7] Rob Miller MIT EECS, "User Interface Design and Implementation", Lecture Notes: <http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-831Fall-2004/LectureNotes/index.htm> [Acedido pela última vez em 25-01-2010].
- [8] Regras básicas para desenhar uma interface de sucesso: http://www.administradores.com.br/home/creativebizz/blog/regras_basicas_para_t_er_um_bom_website/1489/ [Acedido pela última vez em 25-01-2010].
- [9] UML Design - http://docs.kde.org/stable/pt_BR/kdesdk/umbrello/uml-elements.html [Acedido pela última vez em 25-01-2010].
- [10] UML- <http://paginas.fe.up.pt/~jlopes/doc/uml.html> [Acedido pela última vez em 25-01-2010].
- [11] A. M. O. Restivo e J. A. Faria, "Sistemas de informação industriais", Faculdade de Engenharia da Universidade do Porto, 2007-2008.
- [12] PSIM - Programa de simulação para electrónica de potencia, controlo de motores e sistemas dinâmicos - "Powersim"- <http://www.powersimtech.com/feedback.html> [Acedido pela última vez em 25-01-2010].
- [13] NI Multisim- Ferramenta de projectos de circuitos, "National Instruments", <http://www.ni.com/support/> [Acedido pela última vez em 25-01-2010].
- [14] A. Simões, A. Araújo R. Chibante, "Desenvolvimento de programa educativo para o ensino de semicondutores - Semisoft ", Tese de Mestrado, Faculdade de Engenharia da Universidade do Porto, Fevereiro de 2009.
- [15] IEEE Global History Network, "Diode" - <http://www.ieeeahn.org/wiki/index.php/Diode> [Acedido pela última vez em 25-01-2010].
- [16] Wikipedia, "Diode" - <http://en.wikipedia.org/wiki/Diode> [Acedido pela última vez em 25-01-2010].
- [17] William F. Brinkman, Douglas E. Haggan and William W. Troutman, "A History of the Invention of the Transistor and Where It Will Lead Us", IEEE Journal of Solid-State Circuits, Vol.32, December 1997.
- [18] Jonh Worton, "Semiconductors and the Information Revolution", Academic Press, Julho de 2009.
- [19] Jacobus W. Swart, "Evolução de Microelectrônica a Micro-Sistemas", UNICAMP 2000.

- [20] Tudor Jenkins, "A brief history of... semiconductors", Institute of Mathematical and Physical Sciences, University of Wales, 2005.
- [21] IEEE Global History Network, "Edison Effect", http://www.ieeeahn.org/wiki/index.php/Edison_Effect [Acedido pela última vez em 25-01-2010].
- [22] IEEE Global History Network, "Milestones: Fleming Valve, 1904" - http://www.ieeeahn.org/wiki/index.php/Milestones:Fleming_Valve%2C_1904 [Acedido pela última vez em 25-01-2010].
- [23] IEEE Global History Network, "Early Microelectronics" - http://www.ieeeahn.org/wiki/index.php/Early_Microelectronics [Acedido pela última vez em 25-01-2010].
- [24] IEEE Global History Network, "Semiconductors" - <http://www.ieeeahn.org/wiki/index.php/Semiconductors> [Acedido pela última vez em 25-01-2010].
- [25] História do Semicondutor a partir de 1900 - <http://www.tecnosapiens.com.br/2008/09/historia-semicondutor-a-partir-de-1900/> [Acedido pela última vez em 25-01-2010].
- [26] História do Semicondutor antes de 1900 - <http://www.tecnosapiens.com.br/2008/09/historia-semicondutor-antes-de-1900/> [Acedido pela última vez em 25-01-2010].
- [27] Wikipedia, "History of the Transistor" - <http://en.wikipedia.org/wiki/Transistor> [Acedido pela última vez em 25-01-2010].
- [28] Nobelprize.org, "The History of the Integrated Circuit" - http://nobelprize.org/educational_games/physics/integrated_circuit/history/ [Acedido pela última vez em 25-01-2010].
- [29] IEEE Global History Network, "Transistors" - <http://www.ieeeahn.org/wiki/index.php/Transistors> [Acedido pela última vez em 25-01-2010].
- [30] Esquemático da lâmpada de Edison - http://www.feiradeciencias.com.br/sala19/image19/19_72_13.gif [Acedido pela última vez em 25-01-2010].
- [31] Imagem do Efeito de Edison, Válvula de Fleming e Tríodo de Lee de Forrest, http://pt.wikipedia.org/wiki/Efeito_Hall [Acedido pela última vez em 25-01-2010].
- [32] Imagem do primeiro transistor inventado por Bardeen e Brattain, <http://www.inovacaotecnologica.com.br/noticias/imagens/010805071224-transistor1.jpg> [Acedido pela última vez em 25-01-2010].
- [33] Lei de Moore, http://speeches.ofset.org/jrfernandez/2009/sl_y_educac/images/lei_de_moore_2006.png [Acedido pela última vez em 25-01-2010].
- [34] IEEE Global History Network, "Transistor mais fino do mundo é feito com folha de grafeno" [Acedido pela última vez em 25-01-2010]. - <http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=010110070302>
- [35] IEEE Global History Network, "Novo transistor bate recorde mundial de velocidade" <http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=novo-transistor-tritura-recorde-mundial-velocidade&id=010110090911> [Acedido pela última vez em 25-01-2010].
- [36] IEEE Global History Network, "Criado um Díodo molecular" - <http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=diodo-molecular&id=010110091029> [Acedido pela última vez em 25-01-2010].
- [37] IEEE Global History Network, "Nasce o primeiro transistor molecular", <http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=criado-primeiro-transistor-molecular&id=010810091224> [Acedido pela última vez em 25-01-2010].
- [38] Estrutura do grafeno - http://www.lbl.gov/Science-Articles/Archive/sabl/2007/Nov/assets/img/lrg/graphene_sheet.jpg [Acedido pela última vez em 25-01-2010].
- [39] IEEE Global History Network, "Sucessor do silício alcança escala industrial", <http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=grafeno->

- [sucessor-silicio-alcanca-escala-industrial&id=010110100107](#) [Acedido pela última vez em 25-01-2010].
- [40] Lei de Moore - <http://www.intel.com/technology/mooreslaw> [Acedido pela última vez em 25-01-2010].
- [41] Chibante, Rui “Desenvolvimento de um modelo para IGBTs otimizado por um método de base experimental”, Dissertação de doutoramento em Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, Setembro de 2005.
- [42] Borland Delphi 7- <http://delphi.about.com/od/productreviews/a/bld7ann.htm> [Acedido pela última vez em 25-01-2010].
- [43] Comparação entre softwares de design gráfico de aplicações - <http://delphi.about.com/od/devutilities/a/delphivsothers.htm> [Acedido pela última vez em 25-01-2010].
- [44] A.O.Fernandes, M.Kerecki, “Linguagem Object Pascal (Delphi)” http://www.inf.ufsc.br/~adriana/fase_06/ic/INE5622_Cap-3_Ling-Programacao.pdf [Acedido pela última vez em 25-01-2010].
- [45] L.K.Wives, “Introdução à Linguagem de Programação Turbo Pascal” <http://www.leandro.wives.nom.br/pascal/Apostila/Apostila.pdf> [Acedido pela última vez em 25-01-2010].
- [46] MacTech, “Object Pascal” <http://www.mactech.com/articles/mactech/Vol.02/02.12/ObjectPascal/> [Acedido pela última vez em 25-01-2010].
- [47] Inno Setup - <http://www.software.org/isinfo.php> [Acedido pela última vez em 05-02-2010].
- [48] Quickpdf - <http://www.quickpdflibrary.com/> [Acedido pela última vez em 05-02-2010].

Anexo A

1- Ficheiro com os gráficos bidimensionais e respectivas explicações teóricas.

Gráfico 2D

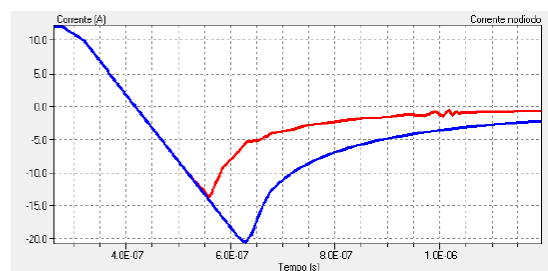
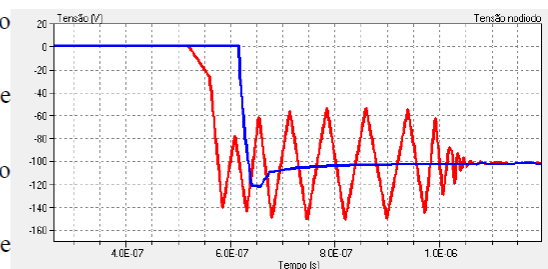
Da teoria dos díodos é de esperar que o aumento do tempo médio de vida provoque

um aumento do tempo de recuperação inversa. Este fenómeno acontece porque, ao aumentar

o tempo médio de vida de um diodo, há um acréscimo de carga disponível na zona N-, que por sua

vez incrementa o tempo de recombinação entre electrões e lacunas provocando assim um aumento

do tempo de recuperação inversa.



- 2- Ficheiro com o circuito e respectiva explicação teórica.

Circuito

Com a entrada em condução do transistor (T1), estabelece-se um valor bem definido de

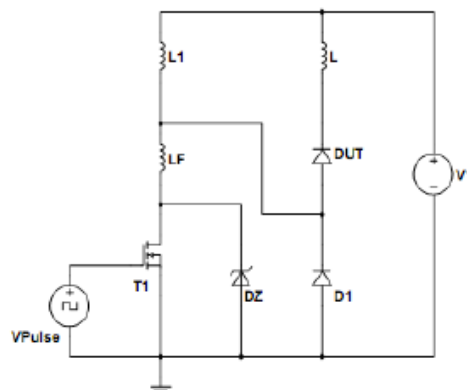
corrente no conjunto de bobines (L1 e LF).

A abertura do transistor (T1), impõe a condução do diodo de teste (DUT) sobre o conjunto de

bobines (L1 e L).

Após o estabelecimento da corrente, um novo fecho do transistor (T1) impõe uma descida da

corrente no diodo (DUT) com um gradiente de $(-V1)/(L+LF)$.



Deve-se salientar que o valor da bobine L1 deve ser escolhido para que durante o tempo de

comutação, se comporte como uma fonte de corrente constante.

A partir do momento em que durante a recuperação inversa, o gradiente da corrente no diodo

DUT se inverte, o diodo D1 entra em condução.

Assim, durante o tempo de armazenamento, o gradiente da corrente é controlado pela bobine

LF.

A recuperação após o pico de corrente inversa e o restabelecimento do poder de bloqueio do

diodo é feito sobre a bobine L.